

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1728

**Nadzirani pristupi za procjenu
nesigurnosti predikcija dubokih
modela**

Ivan Grubišić

Zagreb, lipanj 2018.

Umjesto ove stranice umetnite izvornik Vašeg rada.

Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.

Nadzirani pristupi za procjenu nesigurnosti predikcija dubokih modela

Procjena nesigurnosti predikcija vrlo je važan sastojak mnogih praktičnih primjena konvolucijskih modela računalnog vida. Do tog cilja možemo doći analizom višeznačnosti podataka, nesigurnosti odluke modela te vjerojatnosti da se podatak nalazi u distribuciji skupa za učenje. U ovom radu razmatramo pristupe koji procjenu nesigurnosti predikcija uče nadzirano, primjenom istih podataka na kojima se uči i promatrani model.

U okviru rada, potrebno je proučiti i ukratko opisati postojeće pristupe za procjenu nesigurnosti predikcija. Uhodati postupke procjene nesigurnosti dubokih konvolucijskih modela temeljene na nadziranom učenju. Validirati hiperparametre te prikazati i ocijeniti ostvarene rezultate na problemu semantičke segmentacije. Predložiti pravce budućeg razvoja. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zahvaljujem prof. dr. sc. Siniši Šegviću na pomoći, savjetima i prijedlozima tijekom studiranja i pogotovo tijekom rada na diplomskom radu. Zahvaljujem i asistentima Ivanu Kreši, Marinu Oršiću i Petri Bevandić na pomoći. Zahvaljujem svojoj obitelji na podršci.

SADRŽAJ

Oznake	viii
1. Uvod	1
2. Osnovni pojmovi	3
2.1. Teorija vjerojatnosti	3
2.1.1. Slučajne varijable i razdiobe	3
2.1.2. Zdužena, uvjetna i marginalna vjerojatnost i osnovna pravila vjerojatnosti	5
2.1.3. Nezavisnost, uvjetna nezavisnost i uvjetna zavisnost	6
2.1.4. Očekivanje, varijanca i kovarijanca	7
2.1.5. Funkcije slučajnih varijabli	8
2.1.6. Primjeri razdioba	10
2.2. Teorija informacije	12
2.3. Optimizacija temeljena na gradijentu	15
2.3.1. Gradijentni spust i još neki algoritmi koje se temelje na njemu	16
2.3.2. Postupci drugog reda	18
3. Statističko zaključivanje	20
3.1. Probabilistički grafički modeli	20
3.2. Procjena parametara i zaključivanje	23
3.2.1. Procjenitelji i točkaste procjene parametara	23
3.2.2. Svojstva i pogreška procjenitelja	24

3.2.3.	Procjenitelj maksimalne izglednosti	24
3.2.4.	Procjenitelj maksimalne aposteriorne vjerojatnosti	25
3.2.5.	Bayesovski procjenitelj i zaključivanje	26
3.3.	Monte Carlo aproksimacija	27
3.4.	Aproksimacija razdioba i aproksimacijsko zaključivanje	27
3.5.	Varijacijsko zaključivanje	28
3.5.1.	Metoda srednjeg polja	29
4.	Nadzirano strojno učenje	31
4.1.	Induktivna pristranost	32
4.2.	Komponente algoritma strojnog učenja	32
4.3.	Kapacitet modela, podnaučenost i prenaučenost	33
4.4.	Rizik i funkcija pogreške	34
4.4.1.	Rizik i empirijski rizik	35
4.4.2.	Strukturni rizik i regularizacija	35
4.5.	Odabir modela	36
4.5.1.	Unakrsna validacija	37
4.5.2.	Bayesovska usporedba modela	37
4.6.	Osnovni zadaci nadziranog učenja	38
4.7.	Primjeri modela: poopćeni linearni modeli	38
5.	Duboko učenje i konvolucijske mreže	41
5.1.	Duboke unaprijedne mreže	42
5.2.	Učenje	44
5.2.1.	Algoritam propagacije pogreške unatrag	45
5.2.2.	Gradijenti nekih osnovnih operacija	46
5.2.3.	Stohastička optimizacija	47
5.2.4.	Inicijalizacija parametara	48

5.2.5.	Problem nekonveksnosti funkcije pogreške	49
5.3.	Regularizacija i poboljšavanje učenja	49
5.3.1.	Kažnjavanje norme težina	50
5.3.2.	Rano zaustavljanje učenja	50
5.3.3.	Generiranje podataka	51
5.3.4.	Isključivanje neurona — dropout	51
5.3.5.	Normalizacija po grupama	51
5.3.6.	Neprijateljski primjeri i regularizacija za postizanje otpornosti na njih	53
5.3.7.	Dijeljenje parametara i dijelova mreže	56
5.3.8.	Pomoćni gubici i preskočne veze	56
5.4.	Konvolucijske mreže	57
5.4.1.	Konvolucija	57
5.4.2.	Konvolucijski sloj	58
5.4.3.	Slojevi sažimanja	63
6.	Procjena nesigurnosti kod dubokih modela	65
6.1.	Aleatorna i epistemička nesigurnost	65
6.1.1.	Izvanrazdiobni primjeri	67
6.2.	Važnost i primjene procjene i razlikovanja nesigurnosti	67
6.3.	Bayesovske neuronske mreže	68
6.3.1.	Varijacijsko zaključivanje kod bayesovskih neuronskih mreža	69
6.4.	Mjere za izražavanje nesigurnosti predikcije	70
6.5.	Razlikovanje aleatorne i epistemičke nesigurnosti	71
6.5.1.	Eksplisitno modeliranje aleatorne nesigurnosti	71
6.5.2.	Međusobna informacija kao mjera epistemičke nesigurnosti	73
6.6.	Primjeri pristupa za procjenu nesigurnosti	74
6.6.1.	Aproksimacija bayesovske neuronske mreže pomoću dropota	74

6.6.2. Prepoznavanje izvanrazdiobnih i krivo klasificiranih primjera na temelju izlaza softmaksa ili logita	77
7. Eksperimenti	79
7.1. Evaluacijske mjere za klasifikaciju	79
7.1.1. Binarna klasifikacija	79
7.1.2. Višeklasna klasifikacija	81
7.1.3. Semantička segmentacija	82
7.2. Procjena i razlikovanje nesigurnosti kod semantičke segmentacije pomoću MC-dropouta	82
7.3. Prepoznavanje izvanrazdiobnih i krivo klasificiranih primjera na temelju izlaza softmaksa ili logita kod klasifikacije slika	86
8. Zaključak	91
Literatura	92

Oznake

Objekti

Varijable se označavaju kosim slovima sa serifima, većina konstanti uspravnim slovima sa serifima, a slučajne varijable kosim slovima bez serifa. Vektori se označavaju malim podebljanim slovima, matrice i višedimenzionalni nizovi (tenzori) velikim podebljanim slovima, a skupovi slovima s udvostručenim linijama. Za svaku vrstu objekta mogu se koristiti i latinska i grčka slova.

a, A, θ	Varijabla (najčešće skalar)
$\mathbf{a}, \boldsymbol{\theta}$	Vektor ili niz (najčešće vektor stupac)
$\mathbf{A}, \boldsymbol{\Theta}$	Matrica ili višedimenzionalni niz
\mathcal{A}	Skup ili multiskup
a, A, θ	Konstanta
$\mathbf{a}, \boldsymbol{\theta}$	Konstanta vektor ili niz
$\mathbf{A}, \boldsymbol{\Theta}$	Konstanta matrica ili višedimenzionalni niz
\mathbb{A}	Konstanta skup
a, A, θ	Slučajna varijabla
$\mathbf{a}, \boldsymbol{\theta}$	Slučajni vektor ili niz
$\mathbf{A}, \boldsymbol{\Theta}$	Slučajna matrica ili višedimenzionalni niz
\mathcal{A}	Slučajni skup ili multiskup
a , riječ	Oznaka koja ne predstavlja matematički objekt

Konstante

$\{\}$	Prazni skup
e	Konstanta za koju vrijedi $\frac{d}{dx}e^x = e^x$
$\mathbf{0}$	Nul-vektor
\mathbf{e}_i	i -ti vektor kanonske baze
$\mathbf{1}$	Zbroj svih vektora kanonske baze
\mathbf{I}, \mathbf{I}_n	Matrica identiteta (s n redaka i stupaca)
$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$	Poznati skup
$\mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$	Skup nenegativnih/pozitivnih realnih brojeva

Definiranje skupova i nizova

$a..b$	Kraći zapis za a, \dots, b
$\{a..b\}$	Skup cijelih brojeva od a do b
$\{f(a) : P(a)\}, \{f(a)\}_{P(a)}$	Skup čiji su elementi definirani preko funkcije f i predikata P
$\{f(a)\}_a$	Skup čiji su elementi definirani preko funkcije f i varijabli a iz implicitno određenog skupa
$\{a_1..a_n\}, \{a_i\}_{i=1..n}$	Skup s n elemenata
$[x_1, \dots, x_n]$	Vektor redak
$[a_i]_i, [a_{i,j}]_{i,j}, [a_{i,j,k}]_{i,j,k}$	Višedimenzionalni niz s implicitnim ili neodređenim brojem elemenata
$[a, b)$	Poluzatvoreni interval

Donji i gornji indeks

U donjem i gornjem indeksu oznake mogu biti oznake drugih matematičkih objekata ili slova ili riječi koje ne predstavljaju matematičke objekte. Redni brojevi elemenata vektora ili višedimenzionalnih nizova se, ako nije određeno drugačije, pišu u donjem indeksu oznake vektora u uglatim zagradama. Npr. i -ti element vektora $\mathbf{a} = [a_1, \dots, a_n]^T$ je $\mathbf{a}_{[i]} = a_i$. Indeksi kod n -dimenzionalnih nizova mogu biti i vektori iz \mathbb{N}^n , ili kombinacije vektora manje dimenzije sa skalarima.

a_d^g	Varijabla s oznakama u donjem i gornjem indeksu
$\mathbf{a}_{[i]}$	i -ti element vektora \mathbf{a}
$\mathbf{a}_{[i_1:i_2]}$	Vektor kojeg čine elementi $\mathbf{a}_{[i_1]}, \mathbf{a}_{[i_1+1]}, \dots, \mathbf{a}_{[i_2]}$
$\mathbf{a}_{[(i_1..i_n)]}$	Vektor kojeg čine elementi $\mathbf{a}_{[i_1]}, \mathbf{a}_{[i_2]}, \dots, \mathbf{a}_{[i_n]}$
$\mathbf{A}_{[i,j]}$	Element i, j matrice \mathbf{A}
$\mathbf{A}_{[i,:]}$	i -ti redak matrice \mathbf{A}
$\mathbf{A}_{[:,i_1:i_2,j]}$	2-D odsječak 3-D niza \mathbf{A}
$\mathbf{A}_{[i]}$	Element $\mathbf{A}_{[i_1], \dots, i_{[n]}}$ n -D niza
$\mathbf{A}_{[i_1:i_2]}$	Podniz $\mathbf{A}_{[i_1[1]:i_2[1], \dots, i_1[n]:i_2[n]]}$ n -D niza
$\mathbf{A}_{[i_1:i_2,:]}$	Podniz $\mathbf{A}_{[i_1[1]:i_2[1], \dots, i_1[n-1]:i_2[n-1], :]}$ n -D niza

Operacije linearne algebre i operacije s nizovima

$\langle \mathbf{a} \mathbf{b} \rangle, \mathbf{a}^\top \mathbf{b}$	Skalarni produkt
$\mathbf{a} \mathbf{b}^\top$	Vanjski produkt
$\mathbf{a} \odot \mathbf{b}$	Umnožak po elementima; Hadamardov produkt
$\mathbf{a} \oslash \mathbf{b}$	Dijeljenje po elementima
$\mathbf{a}^{\odot b}$	Potenciranje po elementima
$\mathbf{A} \mathbf{B}$	Matrično množenje
\mathbf{A}^{-1}	Inverz matrice
\mathbf{A}^\top	Transponiranje
$\text{diag}(\mathbf{a})$	Dijagonalna matrica kojoj dijagonalu čini vektor \mathbf{a}
$\det \mathbf{A}$	Determinanta matrice \mathbf{A}
$\ \mathbf{a}\ _2$	L^2 -norma vektora \mathbf{a}
$\ \mathbf{a}\ _p$	L^p -norma vektora \mathbf{a}
$\ \mathbf{A}\ _p$	Matrična L^p -norma matrice \mathbf{A}
$\ \mathbf{A}\ _F$	Frobeniusova norma matrice \mathbf{A}
$\mathbf{a} \# \mathbf{b}$	Konkatenacija vektora (stupaca) $\mathbf{a} \in \mathbb{R}^n$ i $\mathbf{b} \in \mathbb{R}^m$ u vektor iz \mathbb{R}^{n+m}
$\mathbf{A} \# \mathbf{B}$	Konkatenacija nizova po prvoj dimenziji
$\text{vec}(\mathbf{A})$	Funkcija koja preslikava niz iz $\mathbb{R}^{d_1 \times \dots \times d_n}$ u $\mathbb{R}^{d_1 \dots d_n}$
$\dim(\mathbf{a})$	Dimenzija vektora
$\dim(\mathbf{A})$	Vektor dimenzija niza; $[d_1, \dots, d_n]$ za $\mathbf{A} \in \mathbb{R}^{d_1 \times \dots \times d_n}$

Diferencijalni račun

$\frac{dy}{dx}, \frac{d}{dx} f(x)$	Derivacija $y = f(x)$ po x
$\frac{\partial y}{\partial x}, \frac{\partial}{\partial x} f(x)$	Parcijalna derivacija $y = f(x)$ po x
$\nabla_x y, \nabla_x f(x), \left(\frac{\partial y}{\partial x}\right)^\top$	Gradijent $y = f(x)$ po \mathbf{x}
$\nabla_X y, \nabla_X f(x)$	Gradijent $y = f(x)$ po \mathbf{X}
$\frac{\partial^2 y}{\partial x \partial x^\top}, \mathbf{H}_f(\mathbf{x}), \mathbf{H}$	Hesijan iz $\mathbb{R}^{n \times n}$ za $f: \mathbb{R}^n \rightarrow \mathbb{R}$ i $y = f(\mathbf{x})$
$\frac{\partial y}{\partial \mathbf{x}}, \mathbf{J}_f(\mathbf{x}), \mathbf{J}$	Jakobijeva matrica iz $\mathbb{R}^{m \times n}$ za $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ i $\mathbf{y} = f(\mathbf{x})$

$\int_A f(x) dx, \int_{x \in A} f(x)$	Određeni integral funkcije $f(x)$ po $x \in A$
$\int f(x) dx, \int_x f(x)$	Određeni integral funkcije $f(x)$ po $x \in A$, gdje je A implicitan

Teorija vjerojatnosti

Svakoj slučajnoj varijabli a jednoznačno je dodijeljena jedna razdioba $p(a)$ (ili $P(a)$) i funkcija gustoće vjerojatnosti (koja može biti poopćena funkcija) $p_a(a) = p(a = a)$. $P(A)$ označava vjerojatnost događaja A , a P_a funkciju vjerojatnosti slučajne varijable a . Mogući su i kraći zapisi $p(a)$ i $P(a)$, gdje se po slovu koje označava vrijednost pretpostavlja slučajna varijabla označena istim slovom bez serifa. Mogu se koristiti i druge oznake za funkciju vjerojatnosti ili funkciju gustoće vjerojatnosti.

$(a \mid b = b), (a \mid b)$	Uvjetna slučajna varijabla
(a, b)	Združena slučajna varijabla
$a \perp b$	<i>Slučajne varijable a i b su nezavisne</i>
$a \not\perp b$	<i>Slučajne varijable a i b su zavisne</i>
$a \perp b \mid c$	<i>Slučajne varijable a i b su uvjetno nezavisne uz poznat ishod slučajne varijable c</i>
$a \not\perp b \mid c$	<i>Slučajne varijable a i b su uvjetno zavisne uz poznat ishod slučajne varijable c</i>
p, q	Razdioba ili funkcija gustoće vjerojatnosti
A	Događaj
$\{R(a)\}$	Događaj definiran predikatom slučajne varijable a
$P(\{R(a)\}), P(R(a))$	Vjerojatnost događaja $\{R(a)\}$
$P(a), p(a), \mathcal{D}$	Razdioba slučajne varijable a ; P ako je a diskretna slučajna varijabla, a p ako nije ili ako se ne zna
$P(a = a), P_a(a), P(a)$	Vjerojatnost događaja $\{a = a\}$
$p(a = a), p_a(a), p(a)$	Gustoća vjerojatnosti događaja $\{a = a\}$
$p_{a b}(a), p(a \mid b)$	Gustoća vjerojatnosti događaja $\{a = a \mid b = b\}$
$p_{a,b}(a, b), p(a, b)$	Gustoća vjerojatnosti događaja $\{a = a, b = b\}$
$a \sim q, p(a) = q$	<i>Slučajna varijabla a ima razdiobu q</i>
$a \sim A$	<i>Slučajna varijabla a ima takvu razdiobu da svi elementi (multi)skupa A imaju vjerojatnost proporcionalnu višestrukosti ($\frac{1}{ A }$ za običan skup)</i>

$a \sim q$	a se izvlači iz razdiobe q
$a \sim \mathcal{A}, a \sim p(a)$	a se izvlači iz razdiobe $p(a)$
$\mathbb{E}_{a \sim \mathcal{A}} f(a), \mathbb{E}_{\mathcal{A}} f(a)$	Očekivanje funkcije slučajne varijable a
$\mathbb{D}_{a \sim \mathcal{A}} f(a), \mathbb{D}_{\mathcal{A}} f(a)$	Disperzija (varijanca) funkcije slučajne varijable a
$\text{Cov}(a, b)$	Kovarijanca
$\mathcal{N}(\mu, \sigma^2)$	Normalna razdioba s očekivanjem μ i varijancom σ^2
$\mathcal{U}(\mathcal{A})$	Uniformna razdioba nad skupom \mathcal{A}

Teorija informacije

$I(\mathcal{A})$	Sadržaj informacije događaja \mathcal{A}
$H(a)$	Entropija
$h(a)$	Diferencijalna entropija
$I(a, b)$	Međusobna informacija
$H(a \mid b)$	Uvjetna entropija
$H_b(a)$	Unakrsna entropija
$D_{\text{KL}}(a \parallel b)$	Kullback-Leiblerova divergencija (relativna entropija)

Grafovi

$\text{pa}_G(a)$	Skup čvorova roditelja čvora a u grafu G
$\text{ch}_G(a)$	Skup čvorova djece čvora a u grafu G
$\text{pred}_G(a)$	Skup čvorova prethodnika čvora a u grafu G
$\text{succ}_G(a)$	Skup čvorova nasljednika čvora a u grafu G

Ostale matematičke oznake

$\mathcal{A} \rightarrow \mathcal{B}$	Skup funkcija s domenom \mathcal{A} i kodomenom \mathcal{B}
$f: \mathcal{A} \rightarrow \mathcal{B}$	Funkcija s domenom \mathcal{A} i kodomenom \mathcal{B}
$x \mapsto g(x)$	Definicija funkcije; funkcija koja preslikava x iz domene u $g(x)$ iz kodomene
$f + g$	Zbroj funkcija

fg	Umnožak funkcija
$f * g$	Konvolucija funkcija
$\langle f g\rangle$	Skalarni produkt funkcija
$ A $	Kardinalitet skupa
$\delta(\cdot)$	Diracova delta
$\llbracket \cdot \rrbracket$	Iversonova uglata zagrada; $\llbracket P \rrbracket = \begin{cases} 1, & P \equiv \top \\ 0, & P \equiv \perp \end{cases}$

Fraze

dimenzija vektora	Broj komponenata ili kardinalitet baze vektorskog prostora
n -dimenzionalni vektor	Vektor s dimenzijom n
i -ta komponenta vektora \mathbf{a}	$\mathbf{a}_{[i]}$
n -dimenzionalni niz	Niz (engl. <i>array</i>) iz $\mathbb{R}^{d_1 \times \cdots \times d_n}$, tj. postoji $f: \{1..d_1\} \times \cdots \times \{1..d_n\} \rightarrow \mathbb{R}$ tako da za svaku n -torku \mathbf{i} iz njene domene vrijedi $\mathbf{A}_{[i]} = f(\mathbf{i})$
i -ta dimenzija niza	d_i , ako je niz iz $\mathbb{R}^{d_1 \times \cdots \times d_n}$
n -D	n -dimenzionalan

1. Uvod

U mnogim praktičnim primjenama sve složenijih modela strojnog učenja procjena nesigurnosti ima veliku važnost. Osim samog iznosa nesigurnosti, kod nekih zadataka je korisno znati i je li uzrok nesigurnosti višeznačnost podatka ili nedovoljna informiranost koja se može smanjiti uz unošenje više informacija (podataka) u postupak učenja. U ovom radu se razmatraju nadzirani pristupi za procjenu nesigurnosti kod dubokih nadziranih modela za računalni vid. Ti pristupi se mogu podjeliti u dvije skupine. Jednu skupinu čine pristupi koji se temelje na ideji bayesovske procjene parametara modela i omogućuju razlikovanje navedenih uzroka nesigurnosti, a drugu pristupi za prepoznavanje primjera koji su izvan razdiobe skupa za učenje na temelju izlaza uobičajenih klasifikacijskih modela.

U poglavlju 2 su definirani i kratko objašnjeni neki od osnovnih pojmova u vezi teorije vjerojatnosti, teorije informacije i optimizacije temeljene na gradijentu, na kojima se temelje pojmovi u poglavljima koja slijede. U njemu su definirane i neke oznake, a na početku rada je tablični pregled mnogih oznaka koje se koriste u radu. U poglavlju 3 su opisani neki od osnovnih pojmova u vezi statističkog zaključivanja, koje je jako bitno u strojnom učenju. U poglavlju 4 su opisani neki od osnovnih pojmova strojnog učenja, posebno u vezi nadziranog učenja. U poglavlju 5 su neki od osnovnih pojmova dubokog učenja, posebno u vezi unaprijednih neuronskih mreža i konvolucijskih mreža. Poglavlje 6 se bavi glavnom temom ovog rada — procjenom nesigurnosti kod dubokih nadziranih modela. U njemu je opisana podjela nesigurnosti, bayesovske neuronske mreže, mjere za izražavanje nesigurnosti i neki primjeri pristupa za procjenu nesigurnosti ili prepoznavanje nalazi li se podatak u razdiobi skupa za učenje. Neki od tih pristupa ([Kendall i Gal, 2017](#); [Hendrycks i Gimpel, 2016](#); [Liang et al., 2017](#)) su isprobani za ovaj rad. Rezultati eksperimenata su opisani u poglavlju 7.

Rad je pisan u \LaTeX -u. Za slike u radu su korišteni TikZ, Matplotlib (i druge biblioteke u Pythonu) i Inksapes. Za programsku potpuru korišten je programski

jezik Python i biblioteke TensorFlow, NumPy, PyTorch, Scikit-image, Scikit-learn, Matplotlib, SciPy i druge. Izvorni kod za sve je u repozitoriju <https://github.com/Ivan1248/deep-learning-uncertainty>.

2. Osnovni pojmovi

U ovom poglavlju su definirani i kratko objašnjeni neki od osnovnih pojmova na kojima se temelje pojmovi u poglavljima koja slijede.

2.1. Teorija vjerojatnosti

Jako važan pojam u strojnom učenju je nesigurnost ili neizvjesnost. Ona dolazi od šuma u mjerenju i iz konačnosti skupa podataka (Bishop, 2006). Teorija vjerojatnosti nam omogućuje modeliranje nesigurnosti i pronalaženje optimalnih zaključaka korištenjem dostupnih informacija.

Postoje dvije glavne interpretacije vjerojatnosti (Murphy, 2012). Jedna je *frekventistička* interpretacija, prema kojoj vjerojatnosti predstavljaju učestalosti različitih događaja ako se pokus ponavlja velik broj puta. Druga je *bayesovska* interpretacija, prema kojoj vjerojatnost izražava našu nesigurnost o ishodu pokusa.

Ovo poglavlje daje kratak i matematički ne potpuno precizan pregled nekih od osnovnih pojmova i pravila vezanih uz vjerojatnost. Na strukturu ovog poglavlja imaju utjecaj Goodfellow et al. (2016); Murphy (2012).

2.1.1. Slučajne varijable i razdiobe

Neizvjesnost neke pojave modeliramo **slučajnom varijablom**. Slučajnoj varijabli je dodijeljena **razdioba** koja definira skup vrijednosti koje slučajna varijabla može poprimiti i vjerojatnosti ostvarivanja tih vrijednosti. Skup mogućih vrijednosti neke slučajne varijable se još naziva i **prostor elementarnih događaja**. **Elementarni događaj** je element prostora elementarnih događaja i, ako je x slučajna varijabla za koju se u nekom eksperimentu opaža vrijednost x , taj događaj ima zapis $\{x = x\}$, a njegova vjerojatnost se označava $P(\{x = x\})$, $P(x = x)$ ili $P(x)$. **Događaj** je skup

vrijednosti i obično se izražava predikatom nad slučajnom varijablom:

$\{R(x)\} = \{x: R(x)\}$. Ako je \mathbb{X} prostor elementarnih događaja slučajne varijable x , onda $P(x \in \mathbb{X}) = 1$. Funkcija

$$P_x: \mathbb{X} \longrightarrow [0, 1]$$

$$x \longmapsto P(x = x)$$

je **funkcija vjerojatnosti** (engl. *probability mass function, pmf*).

Razlikujemo diskretne i kontinuirane slučajne varijable. Prostor elementarnih događaja diskretne slučajne varijable je prebrojiv skup. Razdioba kontinuirane slučajne varijable x koja poprima vrijednosti iz skupa \mathbb{X} je određena **funkcijom gustoće vjerojatnosti** (engl. *probability density function, pdf*)

$$p_x: \mathbb{X} \longrightarrow [0, \infty)$$

$$x \longmapsto p(x)$$

za koju vrijedi

$$P(x \in A) = \int_A p_x(x) dx \quad (2.1)$$

za svaki $A \subset \mathbb{X}$.

Funkciju gustoće vjerojatnosti možemo smatrati i **poopćenom funkcijom**¹, tj. funkcijom koja se može izraziti kao zbroj neke funkcije f i translahiranih Diracovih funkcija oblika $x \mapsto f(x) + \sum_i \delta(x - \alpha_i)$, gdje su α_i neke konstante, a δ Diracova funkcija, poopćena funkcija za koju vrijedi $\delta(x) = 0$ za $x \neq 0$ i $\int_x \delta(x) dx = 1$. To nam omogućuje da funkcijom gustoće predstavljamo razdiobe za koje neki elementarni događaji imaju vjerojatnost veću od 0. Razdiobu diskretne slučajne varijable x onda možemo predstaviti funkcijom gustoće vjerojatnosti

$$p_x(x) = \sum_{x' \in \mathbb{X}} P(x = x') \delta(x - x'), \quad (2.2)$$

gdje je \mathbb{X} prostor elementarnih događaja slučajne varijable x . Diracova funkcija se može promatrati kao limes funkcije gustoće Gaussove razdiobe:

$$\delta(x) = \lim_{\sigma \rightarrow 0} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

Ako je argument Diracove funkcije vektor $x = (x_1, \dots, x_n)$, onda Diracovu

¹[https://en.wikipedia.org/wiki/Distribution_\(mathematics\)](https://en.wikipedia.org/wiki/Distribution_(mathematics))

funkciju definiramo umnoškom Diracovih funkcija njegovih elemenata:

$$\delta(\mathbf{x}) := \prod_i \delta(x_i). \quad (2.3)$$

Takva definicija omogućuje da n -struki integrali funkcija definiranih izrazima (2.2) ili (2.3) imaju vrijednost 1.

Razdioba slučajne varijable x će se u ovom radu označavati s $P(x)$ ako je diskretna, a s $p(x)$ ako je kontinuirana ili ju nismo definirali. Funkcija (gustoće) vjerojatnosti će se označavati bez oznake slučajne varijable u indeksu ako je po slovu vrijednosti jasno o kojoj se varijabli radi. Druge oznake koje se koriste opisane su u popisu oznaka na početku rada. Na nekim mjestima će, radi kratkoće, riječ *razdioba* imati značenje *funkcija gustoće* ili *funkcija vjerojatnosti*.

2.1.2. Združena, uvjetna i marginalna vjerojatnost i osnovna pravila vjerojatnosti

Dvije razdiobe su iste ako imaju iste funkcije gustoće vjerojatnosti. Dvije slučajne varijable koje imaju istu razdiobu ne moraju biti iste jer se mogu razlikovati po odnosima s drugim slučajnim varijablama.

Možemo razmatrati više slučajnih varijabli zajedno (združenu slučajnu varijablu ili slučajni vektor) i njihovu **združenu razdiobu**, npr. $p(x, y)$. Događaji onda imaju oblik $\{R(x, y)\}$, gdje je R neki predikat. Elementarni događaj onda ima oblik $\{x = x, y = y\}$. Često ćemo $\{x = x, y = y\}$ skraćeno označavati s x, y ako je jasno po slovima o kojim se slučajnim varijablama radi. U ovom odjeljku će pravila vjerojatnosti biti opisana za elementarne događaje, ali ista pravila vrijede i za općenitije događaje jer za svaki događaj kojemu možemo definirati indikatorsku slučajnu varijablu kojoj je taj događaj elementarni događaj.

Uvjetna vjerojatnost je vjerojatnost nekog događaja ako je poznato da se neki drugi događaj ostvario. Ovako je definirana uvjetna vjerojatnost događaja $\{x = x\}$ ako je poznato da se ostvario događaj $\{y = y\}$:

$$p(x | y) := \frac{p(x, y)}{p(y)}. \quad (2.4)$$

Združena vjerojatnost se može rastaviti **pravilom umnoška**:

$$p(x, y) = p(x | y) p(y). \quad (2.5)$$

Općenitije, pravilo umnoška za n slučajnih varijabli x_1, \dots, x_n izgleda ovako:

$$p(x_1, \dots, x_n) = p(x_1) p(x_2 | x_1) \cdots p(x_n | x_1, \dots, x_{n-1}) \quad (2.6)$$

$$= p(x_1) \prod_{i=2..n} p(x_i | x_1, \dots, x_{i-1}). \quad (2.7)$$

Marginalna vjerojatnost slučajne varijable x je $p(x) = p(x = x, y \in \mathbb{Y})$, gdje je \mathbb{Y} prostor elementarnih događaja slučajne varijable y . Izraženo gustoćom vjerojatnosti (**pravilo zbroja, marginalizacija**):

$$p(x) = \int_{\mathbb{Y}} p(x, y) dy = \int_{\mathbb{Y}} p(x | y) p(y) dy. \quad (2.8)$$

Dvije slučajne varijable koje imaju istu razdiobu ne moraju biti u istom odnosu prema drugim slučajnim varijablama. Npr. ako $x_1 \sim q_1$, $x_2 \sim q_1$ i $y \sim q_2$, tj. slučajne varijable x_1 i x_2 imaju razdiobu q_1 , a y razdiobu q_2 , ne mora vrijediti $p(x_1, y) = p(x_2, y)$.

Rastavljanjem lijeve strane jednadžbe (2.5) na umnožak $p(x | y) p(y)$ dobivamo **Bayesovo pravilo**:

$$p(x | y) = \frac{p(y | x) p(x)}{p(y)}, \quad (2.9)$$

što možemo i ovako zapisati:

$$p(x | y) = \frac{p(y | x) p(x)}{\int p(y | x) p(x) dx}, \quad (2.10)$$

gdje se nazivnik integrira po svim vrijednostima.

2.1.3. Nezavisnost, uvjetna nezavisnost i uvjetna zavisnost

Kada su dvije slučajne varijable x i y **zavisne**, što se označava $x \not\perp y$, znanje o ishodu jedne utječe na znanje o ishodu druge, tj. uvjetna razdioba $p(x | y = y)$ ovisi o ishodu y . *Znanje o ishodu* ne mora značiti da je ishod poznat. Dovoljna je promjena znanja o razdiobi koja može biti posljedica opažanja neke treće slučajne varijable. Slučajne varijable x i y su **nezavisne**, što se označava $x \perp y$, akko za

svaki par (x, y) vrijedi

$$p(x, y) = p(x) p(y), \quad (2.11)$$

ili, ekvivalentno,

$$p(x | y) = p(x). \quad (2.12)$$

Znanje o ishodu jedne slučajne varijable onda ne utječe na znanje o ishodu druge.

Slučajne varijable x i y , koje mogu biti zavisne, su uz znanje o ishodu slučajne varijable z **uvjetno nezavisne**, što se označava $x \perp y | z$, akko su slučajne varijable $(x | z = z)$ i $(y | z = z)$ nezavisne za svaki mogući ishod z . Onda za svaku trojku (x, y, z) vrijedi

$$p(x, y | z) = p(x | z) p(y | z), \quad (2.13)$$

ili, ekvivalentno,

$$p(x | y, z) = p(x | z). \quad (2.14)$$

Isto tako, slučajne varijable x i y koje su nezavisne mogu biti **uvjetno zavisne** uz znanje o ishodu neke slučajne varijable z . Općenito, dvije slučajne varijable ne moraju biti ni uvjetno zavisne ni uvjetno nezavisne jer uz neke ishode treće slučajne varijable po kojoj se uvjetuje one mogu biti zavisne, a uz neke nezavisne. Također se može govoriti i o zavisnosti ili nezavisnosti događaja.

2.1.4. Očekivanje, varijanca i kovarijanca

Očekivanje (prvi moment) slučajne varijable definirano je ovako:

$$\mathbf{E} x := \int x p(x) dx, \quad (2.15)$$

gdje se integrira po prostoru elementarnih događaja. Još se označava ovako: μ_x .

Očekivanje funkcije slučajne varijable zapisujemo ovako:

$$\mathbf{E}_{x \sim x} f(x) := \mathbf{E} f(x) = \int f(x) p(x) dx. \quad (2.16)$$

Ako je po oznaci jasno o kojoj se slučajnoj varijabli radi, možemo kraće pisati $\mathbf{E}_x f(x)$. Očekivanje ima svojstvo linearnosti:

$$\mathbf{E}(\alpha f(x) + \beta g(x)) = \alpha \mathbf{E} f(x) + \beta \mathbf{E} g(x). \quad (2.17)$$

Varijanca (dispersija, drugi centralni moment) slučajne varijable definirana je ovako:

$$\mathbf{D} x := \mathbf{E}[(x - \mathbf{E} x)^2] = \int (x - \mathbf{E} x)^2 p(x) dx. \quad (2.18)$$

Varijanca se može izraziti preko drugog momenta $\mathbf{E} x^2$ i kvadrata očekivanja $(\mathbf{E} x)^2$:

$$\mathbf{D} x = \mathbf{E}((x - \mathbf{E} x)^2) \quad (2.19)$$

$$= \mathbf{E}(x^2 - 2x \mathbf{E} x + (\mathbf{E} x)^2) \quad (2.20)$$

$$= \mathbf{E} x^2 - 2(\mathbf{E} x)^2 + (\mathbf{E} x)^2 \quad (2.21)$$

$$= \mathbf{E} x^2 - (\mathbf{E} x)^2. \quad (2.22)$$

Drugi korijen varijance je standardna devijacija σ_x .

Kovarijanca para slučajnih varijabli definirana je ovako:

$$\text{Cov}(x, y) := \mathbf{E}[(x - \mathbf{E} x)(y - \mathbf{E} y)] = \mathbf{E} xy - (\mathbf{E} x)(\mathbf{E} y). \quad (2.23)$$

Kovarijacijska matrica slučajnog vektora $\mathbf{x} \in \mathbb{R}^n$ je matrica tipa $n \times n$ takva da:

$$\text{Cov}(\mathbf{x})_{[i,j]} = \text{Cov}(x_{[i]}, x_{[j]}). \quad (2.24)$$

Dijagonalni elementi te matrice su $\text{Cov}(\mathbf{x})_{[i,i]} = \mathbf{D} x_{[i]}$.

2.1.5. Funkcije slučajnih varijabli

Neka je odnos između slučajnih varijabli x i y definiran funkcijom f koja ishode jedne slučajne varijable deterministički preslikava u ishode druge, što se označava ovako: $y = f(x)$. Ako su x i y diskretne slučajne varijable, onda je razdioba slučajne varijable y definirana ovako:

$$P_y(y) = \sum_{x: f(x)=y} P_x(x). \quad (2.25)$$

Ako su x i y kontinuirane slučajne varijable s vrijednostima iz \mathbb{R} i f je injektivna, može se pokazati (Elezović, 2007) da vrijedi

$$p_y(y) = p_x(x) \left| \frac{dx}{dy} \right|. \quad (2.26)$$

Neka je $C_x(x) := \int_{-\infty}^x p_x(x') dx'$. Vrijednosti iz intervala $(x, x + \epsilon)$ na kojem je f monotono rastuća preslikavaju se u interval $(f(x), f(x + \epsilon))$. Granice su obrnute ako je f monotono padajuća na tom intervalu. Budući da događaji $\{x \in (x, x + \epsilon)\}$ i $\{y \in (f(x), f(x + \epsilon))\}$ imaju istu vjerojatnost, vrijedi

$$C_x(x + \epsilon) - C_x(x) = C_y(f(x + \epsilon)) - C_y(f(x)). \quad (2.27)$$

Ako obje strane jednadžbe dijelimo s ϵ i pustimo $\epsilon \rightarrow 0$,

$$\lim_{\epsilon \rightarrow 0} \frac{C_x(x + \epsilon) - C_x(x)}{\epsilon} = \lim_{\epsilon \rightarrow 0} \frac{C_y(f(x + \epsilon)) - C_y(f(x))}{\epsilon}. \quad (2.28)$$

Redom prema definiciji derivacije, pravilu derivacije složene funkcije i definiciji funkcija C_x i C_y kao integrala gustoće vjerojatnosti slijedi:

$$\frac{d}{dx} C_x(x) = \frac{d}{dx} C_y(f(x)), \quad (2.29)$$

$$\frac{d}{dx} C_x(x) = \frac{d}{df(x)} C_y(f(x)) \frac{d}{dx} f(x), \quad (2.30)$$

$$p_x(x) = p_y(f(x)) \frac{d}{dx} f(x). \quad (2.31)$$

Može se pokazati da je za monotono padajuće intervale desna strana jednadžbe (2.31) pomnožena s -1 , iz čega uz jednadžbu (2.31) slijedi

$$p_x(x) = p_y(y) \left| \frac{dy}{dx} \right|, \quad (2.32)$$

gdje je $f(x)$ zamijenjen s y . Množenjem toga s $\left| \frac{dx}{dy} \right| = \left| \frac{dy}{dx} \right|^{-1}$ slijedi jednadžba (2.26). To pravilo se može poopćiti i na vektore. Onda je $p_y(\mathbf{y}) = p_x(\mathbf{x}) \left| \det \frac{\partial \mathbf{x}}{\partial \mathbf{y}} \right|$ (Murphy, 2012).

Neka je z zbroj slučajnih varijabli x i y . Onda vrijedi

$$p_z(z) = \int p_{x,y}(x, z - x) dx. \quad (2.33)$$

Ako su x i y nezavisne, onda to postaje konvolucija:

$$p_z(z) = \int p_x(x)p_y(z-x) dx =: (p_x * p_y)(z). \quad (2.34)$$

2.1.6. Primjeri razdioba

Bernoullijeva razdioba je binarna razdioba s prostorom elementarnih događaja koji je obično $\{0, 1\}$. Ona je onda određena parametrom $\mu \in [0, 1]$ i ima ova svojstva:

$$P(x) = \mu \mathbb{I}[x = 1] + (1 - \mu) \mathbb{I}[x = 0] = \mu^x (1 - \mu)^{1-x}, \quad (2.35)$$

$$\mathbf{E} x = \mu, \quad (2.36)$$

$$\mathbf{D} x = \mu(1 - \mu). \quad (2.37)$$

Kategorička razdioba je poopćenje Bernoullijeve razdiobe na konačan prostor elementarnih događaja koji može imati više od 2 vrijednosti. Ako prostor elementarnih događaja ima kardinalitet n , razdioba je određena vektorom $\mathbf{p} \in [0, 1]^{n-1}$ za koji vrijedi $\sum_i p_{[i]} \leq 1$. Prostor elementarnih događaja ne mora biti skup $\{1..n\}$, pa je kategorička razdioba najopćenitija diskretna razdioba nad konačnim skupom elementarnih događaja.

Eksponecijalna razdioba je kontinuirana razdioba s domenom $\mathbb{R}_{\geq 0}$. Ona je definirana parametrom $\lambda \in \mathbb{R}_{>0}$ ili $\beta = \lambda^{-1}$ i ima ova svojstva:

$$p(x) = \lambda \exp(-\lambda x) \quad (2.38)$$

$$\mathbf{E} x = \lambda^{-1}, \quad (2.39)$$

$$\mathbf{D} x = \lambda^{-2}. \quad (2.40)$$

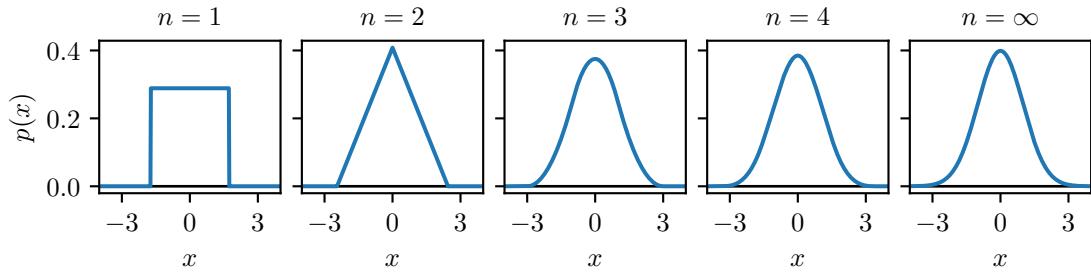
Laplaceova razdioba je kontinuirana razdioba definirana parametrima $\beta \in \mathbb{R}_{>0}$ i $\mu \in \mathbb{R}$ i ima ova svojstva:

$$p(x) = \frac{1}{2\beta} \exp\left(-\frac{|x|}{\beta}\right) \quad (2.41)$$

$$\mathbf{E} x = \mu, \quad (2.42)$$

$$\mathbf{D} x = \beta^2. \quad (2.43)$$

Gaussova (normalna) razdioba $\mathcal{N}(\mu, \sigma^2)$ je kontinuirana razdioba definirana



Slika 2.1: Ilustracija centralnog graničnog teorema. Grafovi za različite brojeve pribrojnika n prikazuju funkcije gustoće vjerojatnosti normaliziranih zbrojeva (kao u je jednadžbi (2.47)) nezavisnih slučajnih varijabli s razdiobom prikazanom prvim grafom. Zadnji graf prikazuje funkciju gustoće Gaussove razdiobe s očekivanjem 0 i varijancom 1. Slika je dobivena ponavljanom konvolucijom (jednadžba (2.34)) funkcije gustoće vjerojatnosti uz primjenu jednadžbe (2.26) za normalizaciju.

parametrima $\mu \in \mathbb{R}$ i $\sigma \in \mathbb{R}_{>0}$ i ima ova svojstva:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.44)$$

$$\mathbf{E} x = \mu, \quad (2.45)$$

$$\mathbf{D} x = \sigma^2. \quad (2.46)$$

Neka je

$$z_n = \frac{\sum_{i=1}^n (x_i - \mu)}{\sigma\sqrt{n}} \quad (2.47)$$

normalizirani zbroj n nezavisnih slučajnih varijabli x_i koje imaju jednaku razdiobu s očekivanjem μ i varijancom σ^2 . Prema **centralnom graničnom teoremu**, z_n u razdiobi konvergira prema Gaussovoj razdiobi kada $n \rightarrow \infty$, tj.

$$\lim_{n \rightarrow \infty} P(z_n < z) = \int_{-\infty}^z p_{\mathcal{N}(0,1)}(z') dz'. \quad (2.48)$$

$p_{\mathcal{N}(0,1)}$ označava funkciju gustoće normalne razdiobe s $\mu = 0$ i $\sigma^2 = 1$. To je detaljnije objašnjeno i dokazano npr. u [Elezović \(2007\)](#). Centralni granični teorem je ilustriran na slici 2.1.

Višedimenzionalna Gaussova razdioba je kontinuirana razdioba definirana

parametrima $\boldsymbol{\mu} \in \mathbb{R}^n$ i pozitivno definitnom matricom $\boldsymbol{\Sigma}$ i ima ova svojstva:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (2.49)$$

$$\mathbf{E} \mathbf{x} = \boldsymbol{\mu}, \quad (2.50)$$

$$\text{Cov}(\mathbf{x}) = \boldsymbol{\Sigma}. \quad (2.51)$$

Ako su $\mathbf{x}_{[i]}$ nezavisne, $\boldsymbol{\Sigma}$ će biti dijagonalna matrica, ali mora vrijediti obrnuto.

2.2. Teorija informacije

Jedan od osnovnih pojmova u teoriji informacije (Shannon, 1948) je **sadržaj informacije** koji događaj preslikava u nenegativan realni broj:

$$I(x \in A) := \log_b \frac{1}{P(x \in A)} = -\log_b P(x \in A). \quad (2.52)$$

Događaji koji imaju manju vjerojatnost sadrže više informacije. Ako je vjerojatnost nekog događaja 1, njegov sadržaj informacije je 0. b je najčešće 2 ili e .

Sadržaj informacije odgovara minimalnom broju simbola (bitova ako $b = 2$) potrebnih za kodiranje elementarnih događaja prefiksni kodom za koji je očekivanje duljine poruke minimalno (Olah, 2015b). Kod prefiksnog koda nijedna kodna riječ nije prefiks neke druge kodne riječi. Takav kod se može prenositi kao niz združenih kodnih riječi bez posebnog simbola za označavanje granica između kodnih riječi. Donja granica očekivanja duljine poruke kod optimalnog koda naziva se **entropija**:

$$H(\mathbf{x}) := \mathbf{E}_x I(\mathbf{x} = x) = -\mathbf{E}_x \log_b P(x). \quad (2.53)$$

Kažemo *donja granica* zato što kodne riječi ili poruke mogu imati samo cjelobrojne duljine, a sadržaj informacije pojedinih poruka ne mora biti cjelobrojan, pa optimalno kodirana poruka mora imati najbliži cijeli broj simbola koji je veći ili jednak sadržaju informacije. U nastavku se, radi jednostavnosti i na račun preciznosti, to neće spominjati kod opisivanje drugih informacijsko-teorijskih veličina. Entropija iskazuje neizvjesnost diskretne slučajne varijable. Ona će biti 0 ako je vjerojatnost nekog elementarnog događaja 1, a najveća će biti kada svi elementarni događaji imaju istu vjerojatnost: $H(\mathbf{x}) = \log_b n$, gdje je n broj elementarnih događaja.

Entropija kontinuirane slučajne varijable je beskonačna. Ako se u izrazu (2.53) vjerojatnost zamijeni gustoćom vjerojatnosti, onda on predstavlja **diferencijalnu**

$H(x)$	$D_{KL}(x \parallel y)$
$H_y(x)$	

Slika 2.2: Odnos entropije, unakrsne entropije i KL-divergencije:

$$H_y(x) = H(x) + D_{KL}(x \parallel y).$$

entropiju

$$h(x) := -\mathbf{E}_x \log_b p(x), \quad (2.54)$$

jedan od analoga² entropije za kontinuirane varijable koji nema neka od svojstava koja ima entropija.

Unakrsna entropija je mjera koja iskazuje očekivanje duljine poruke kodirane optimalnim kodom za razdiobu $P(y)$ ako izvor poruka ima razdiobu $P(x)$. Ovako je definirana:

$$H_y(x) := \mathbf{E}_x I(y = x) = -\mathbf{E}_x \log_b P_y(x). \quad (2.55)$$

Za $P(y) = P(x)$ je $H_y(x) = H_x(x) = H(x)$. Za unakrsnu entropiju se često koristi oznaka $H(x, y)$, ali ista oznaka se koristi i za entropiju para slučajnih varijabli (x, y) . Po uzoru na [Olah \(2015b\)](#), ovdje koristimo oznaku $H_y(x)$.

Kao mjera razlike između dviju razdioba često se koristi **relativna entropija** ili **Kullback-Leiblerova divergencija** (KL-divergencija):

$$D_{KL}(x \parallel y) := H_y(x) - H(x) = \mathbf{E}_x \log_b \frac{P_x(x)}{P_y(x)}. \quad (2.56)$$

Ona je uvijek pozitivna i mjeri koliko simbola više se u prosjeku koristi ako se opaža razdioba $P(x)$, a događaji se kodiraju kodom optimalnim za razdiobu $P(y)$.

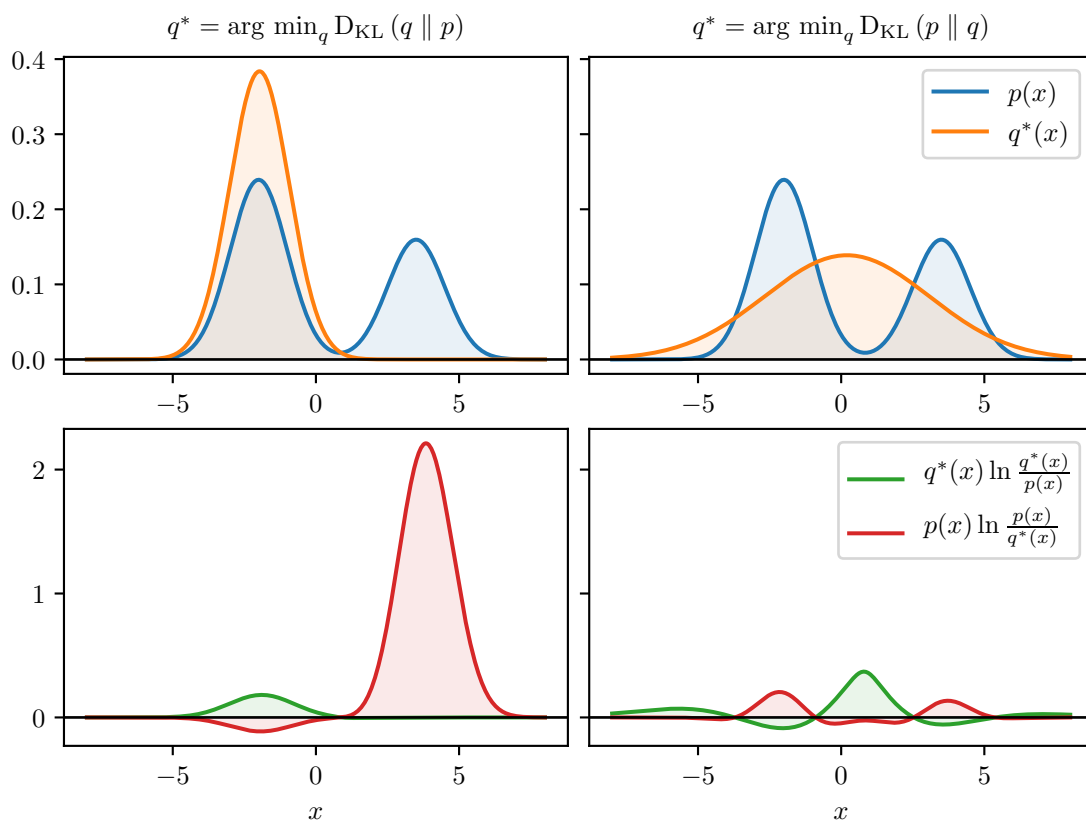
KL-divergencija će biti 0 akko x i y imaju iste razdiobe. To je ilustrirano slikom 2.2.

KL-divergencija, kao ni unakrsna entropija, nije simetrična (slika 2.3), tj. općenito $D_{KL}(x \parallel y) \neq D_{KL}(y \parallel x)$ i $H_y(x) \neq H_x(y)$. KL-divergencija je izrazom (2.56)

definirana i za kontinuirane slučajne varijable ako se funkcije vjerojatnosti zamijene funkcijama gustoće vjerojatnosti. Ona divergira kada postoji x za koji $P_x(x) > 0$ i $P_y(x) = 0$ ili, u slučaju kontinuiranih razdioba, $p_x(x) > 0$ i $p_y(x) = 0$.

Međusobna informacija je mjera zavisnosti između slučajnih varijabli.

²https://en.wikipedia.org/wiki/Differential_entropy



Slika 2.3: Asimetričnost KL-divergencije. p je fiksna razdioba (funkcija gustoće), a q^* je Gaussova razdioba koja ju aproksimira minimizacijom KL-divergencije $D_{KL}(q \parallel p)$ (lijevo) ili $D_{KL}(p \parallel q)$ (desno). U donjem retku grafovi prikazuju podintegralne funkcije odgovarajućih KL-divergencija. Kod njih zbrojevi predznačenih površina obojanih područja odgovaraju KL-divergencijama $D_{KL}(q \parallel p)$ (zeleno) ili $D_{KL}(p \parallel q)$ (crveno). Optimalna aproksimirajuća razdioba desno ima veliku gustoću gdje god razdioba p ima veliku gustoću. Lijevo optimalna aproksimirajuća razdioba nema veliku gustoću gdje razdioba p nema veliku gustoću. Da je razmak između komponenata razdiobe p malo manji, i lijeva razdioba q^* bi pokrila oba moda i bila sličnija desnoj. Slika je napravljena po uzoru na sliku 3.6 u Goodfellow et al. (2016).

H(x)		
H(x y)	I(x, y)	H(y x)
	H(y)	
H(x, y)		

Slika 2.4: Odnosi informacijsko-teorijskih veličina dviju slučajnih varijabli, prema jednačbama (2.58)-(2.60).

Definirana je ovako:

$$I(x; y) := \mathbf{E}_{x,y} \log_b \frac{P_{x,y}(x, y)}{P_x(x)P_y(y)}, \quad (2.57)$$

a može se i na ove načine izraziti:

$$I(x; y) = H(x) + H(y) - H(x, y) \quad (2.58)$$

$$= H(x) - H(x | y) \quad (2.59)$$

$$= H(y) - H(y | x), \quad (2.60)$$

gdje je

$$H(x | y) := \mathbf{E}_x H(y | x = x) \quad (2.61)$$

uvjetna entropija. Ako su x i y nezavisne, njihova međusobna informacija će biti 0. Ako npr. postoji surjekcija f tako da $y = f(x)$, tj. poznavanje ishoda varijable x jednoznačno određuje ishod varijable y , onda $H(y | x) = 0$ i $I(x; y) = H(y)$. Ako je f bijekcija, onda $I(x; y) = H(x) = H(y)$. Definirane veličine mogu se prikazati kao na slici 2.4. Isti odnosi vrijede ako se entropija zamijeni diferencijalnom entropijom.

2.3. Optimizacija temeljena na gradijentu

U ovom odjeljku su opisani osnovni optimizacijski algoritmi temeljeni na gradijentu i neki izvedeni algoritmi koji koriste dodatne heuristike. Oni su bitni u strojnom učenju (poglavlje 4), posebno u dubokom učenju (poglavlje 5). Primjena optimizacijskih algoritama u dubokom učenju opisana je u pododjeljku 5.2.3.

Neka je $f: \mathbb{R}^n \rightarrow \mathbb{R}$ funkcija čiji minumum s obzirom na parametre x želimo naći. Ona se u okolini točke x , ako je dovoljno (beskonačno) puta derivabilna može

izraziti Taylorovim redom:

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \frac{\partial^2 y}{\partial \mathbf{x} \partial \mathbf{x}^\top} \mathbf{x} f(\mathbf{x}) \mathbf{d} + \dots \quad (2.62)$$

S drugačijim oznakama:

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x})^\top \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \mathbf{H}_f(\mathbf{x}) \mathbf{d} + \dots \quad (2.63)$$

2.3.1. Gradijentni spust i još neki algoritmi koje se temelje na njemu

Ako je \mathbf{d} ima malu normu, funkciju f u okoline neke točke možemo dobro aproksimirati s prvih nekoliko članova Taylorovog reda. **Gradijentni spust** je optimizacijski algoritam koji koristi linearnu aproksimaciju i iterativnim ažuriranjem parametara u smjeru gradijenta (*najstrmijem* smjeru) traži minimum. Iteracija gradijentnog spusta ima ovakav oblik:

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_{i-1}), \quad (2.64)$$

gdje je i redni broj iteracije, a η **veličina koraka** (**stopa učenja** kod strojnog učenja) koja može biti konstanta ili može ovisiti o i .

Neka $\mathbf{g} = \nabla_{\mathbf{x}} f(\mathbf{x})$ i $\mathbf{H} = \mathbf{H}_f(\mathbf{x})$. Za dovoljno mali η

$$f(\mathbf{x} - \eta \mathbf{g}) = f(\mathbf{x}) - \eta \mathbf{g}^\top \mathbf{g} - \frac{1}{2} \eta^2 \mathbf{g}^\top \mathbf{H} \mathbf{g} + \dots \quad (2.65)$$

Uz neke blage uvjete koje mora zadovoljavati f i dovoljno mali η , gradijentni spust konvergira, tj. proizvoljno se približi nekom lokalnom minimumu ili stacionarnoj točki koja nije lokalni minimum, u kojoj je isto $\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{0}$, ovisno o η . Jedan blagi uvjet može biti **Lipschitz-kontinuiranost** funkcije f ili njene derivacije (Goodfellow et al., 2016). Funkcija f je Lipschitz-kontinuirana ako postoji konstanta λ za koju za svaki par (\mathbf{x}, \mathbf{y}) vrijedi:

$$|f(\mathbf{x}) - f(\mathbf{y})| < \lambda \|\mathbf{x} - \mathbf{y}\|. \quad (2.66)$$

Najmanji takav λ naziva se **Lipschitzova konstanta**.

Gradijentni spust s inercijom

Jedna heuristika koja je često korisna kod optimizacije funkcija koje su nam zanimljive je simuliranje inercije. Jedan korak **gradijentnog spusta s inercijom** (engl. *momentum gradient descent*) ima ovakav oblik:

$$\mathbf{v}_i = \beta \mathbf{v}_{i-1} + \nabla_{\mathbf{x}} f(\mathbf{x}_{i-1}), \quad (2.67)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta \mathbf{v}_i, \quad (2.68)$$

gdje je $\beta \in [0, 1)$ faktor kojim se određuje *otpor* proporcionalan *brzini* \mathbf{v} , tj. otpor je proporcionalan faktoru $(1 - \beta)$, što se bolje vidi ako se jednačba (2.67) izrazi ovako:

$$\mathbf{v}_i = \mathbf{v}_{i-1} - (1 - \beta) \mathbf{v}_{i-1} + \nabla_{\mathbf{x}} f(\mathbf{x}_{i-1}). \quad (2.69)$$

β se obično odabire da bude bliže 1. Uz $\beta = 1$ dobiva se obični gradijentni spust, a uz $\beta = 0$ čestica koja klizi bez trenja. Uz dobro odabran β prigušuju se pomaci koji nisu u smjeru gibanja \mathbf{v} . To omogućuje bržu konvergenciju i izbjegavanje malih lokalnih optimuma i drugih stacionarnih točaka. Svojstva gradijentnog spusta s inercijom su dobro objašnjena u [Goh \(2017\)](#).

Jedno poboljšanje gradijentnog spusta s inercijom je **Nesterovljev postupak** ([Nesterov, 2014](#); [Sutskever, 2013](#)):

$$\mathbf{v}_i = \beta \mathbf{v}_{i-1} + \nabla_{\mathbf{x}} f(\mathbf{x}_{i-1} - \eta \mathbf{v}_{i-1}), \quad (2.70)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta \mathbf{v}_i. \quad (2.71)$$

Brzina se ažurira uz procjenu gradijenta u budućoj točki na temelju brzine iz prethodne iteracije. Onda se izračuna novi pomak uz tako ažuriranu brzinu.

Primjeri algoritama koji koriste još neke heuristike

Kod opisanih algoritama konvergenciju mogu usporavati područja u kojima gradijent ima male vrijednosti. Način na koji se ta pojava može poništiti je npr. da se gradijent podijeli s njegovom normom. Na taj način će pomaci ovisiti samo o stopi učenja, a ne o strmosti funkcije koja se minimizira. Algoritam **RMSPProp** (opisan npr. u [Hinton \(2012\)](#) ili [Ruder \(2016\)](#)) ostvaruje nešto slično. Iteracija

RMSPropa izgleda ovako:

$$\mathbf{g}_i = \nabla_x f(\mathbf{x}_{i-1}), \quad (2.72)$$

$$\mathbf{r}_i = \rho \mathbf{r}_{i-1} + (1 - \rho) \mathbf{g}_i^{\odot 2}, \quad (2.73)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta (\epsilon \mathbf{1} + \mathbf{r}_i)^{\odot -\frac{1}{2}} \odot \mathbf{g}_i, \quad (2.74)$$

gdje je $\rho \in [0, 1)$ faktor koji određuje koliko se brzo ažurira pokretni prosjek gradijenta kvadriranog po elementima, a ϵ neka mala konstanta. ρ se obično odabire da bude blizu 1. Za $\rho = 0$, ako se ϵ zanemari, dobiva se iteracija algoritma **Rprop** (Hinton, 2012): $\mathbf{x}_i = \mathbf{x}_{i-1} - \text{sgn}(\nabla_x f(\mathbf{x}_{i-1}))$. RMSPropu se još može dodati inercija.

Jedan algoritam koji često ubrzava učenje je **Adam** (Kingma i Ba, 2014). On uključuje i inerciju i skaliranje. Ime Adam izvedeno je iz *adaptive moment estimation*. Jedna iteracija tog algoritma izgleda ovako:

$$\mathbf{g}_i = \nabla_x f(\mathbf{x}_{i-1}), \quad (2.75)$$

$$\mathbf{v}_i = \beta_1 \mathbf{v}_{i-1} + (1 - \beta_1) \mathbf{g}_i, \quad (2.76)$$

$$\mathbf{r}_i = \beta_2 \mathbf{r}_{i-1} + (1 - \beta_2) \mathbf{g}_i^{\odot 2}, \quad (2.77)$$

$$\mathbf{v}'_i = \frac{1}{1 - \beta_1^i} \mathbf{v}_i, \quad (2.78)$$

$$\mathbf{r}'_i = \frac{1}{1 - \beta_2^i} \mathbf{r}_i, \quad (2.79)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta \left(\epsilon \mathbf{1} + \mathbf{r}'_i \right)^{\odot -\frac{1}{2}} \odot \nabla_x f(\mathbf{x}_{i-1}), \quad (2.80)$$

gdje je \mathbf{v}_i pomični prosjek gradijenta, \mathbf{r}_i pokretni prosjek kvadrata gradijenta po elementima, a ϵ mala konstanta. Parametar β_1 ima ulogu kao β kao gradijentnog spusta s inercijom, a β_2 kao ρ kod RMSPropa. Brzina \mathbf{v}_i i pokretni prosjek kvadrata \mathbf{r}_i se inicijaliziraju u $\mathbf{0}$ i u svakom koraku se skaliraju obrnuto proporcionalno udjelu gradijenta u odnosu na inicijalnu vrijednost $\mathbf{0}$ u pokretnom prosjeku radi poništavanja pristranosti procjena. Za faktore skaliranja vrijedi $\lim_{i \rightarrow \infty} \frac{1}{1 - \beta^i} = 1$.

2.3.2. Postupci drugog reda

Ovaj pododjeljak se temelji na Goodfellow et al. (2016, pododjeljak 4.3.1).

Ako koristimo kvadratnu aproksimaciju (2.65), možemo pokušati pronaći optimalni η koji ju minimizira. η za koji $\frac{\partial}{\partial \eta} f(\mathbf{x} - \eta \mathbf{g}) = 0$ će, ako $\mathbf{g}^T \mathbf{H} \mathbf{g} > 0$ dati

minimum u smjeru gradijenta kvadratne aproksimacije funkcije f u točki \mathbf{x} . Dobije se:

$$\eta = \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}}. \quad (2.81)$$

Ako je $f: \mathbb{R}^n \rightarrow \mathbb{R}$ konveksna (pozitivno definitna) kvadratna funkcija, izmijenjeni algoritam gradijentnog spusta, koji ovako određuje veličinu koraka, minimum pronalazi u najviše n koraka.

Postupak drugog reda koji se ne ograničava na pomake u smjeru gradijenta je **Newton-Raphsonov postupak**. Deriviranjem desne strane jednadžbe (2.63) po \mathbf{d} i izjednačavanjem s 0 dobiva se:

$$\mathbf{0} = \nabla_{\mathbf{x}} f(\mathbf{x})^\top + \mathbf{d}^\top \mathbf{H}_f(\mathbf{x}) + \dots. \quad (2.82)$$

Uz kvadratnu aproksimaciju i kraće oznake $\mathbf{g} = \nabla_{\mathbf{x}} f(\mathbf{x})$ i $\mathbf{H} = \mathbf{H}_f(\mathbf{x})$: $\mathbf{H} \mathbf{d} = -\mathbf{g}$. Slijedi da je pomak \mathbf{d} koji daje stacionarnu točku aproksimacije

$$\mathbf{d} = -\mathbf{H}^{-1} \mathbf{g}. \quad (2.83)$$

Za nekvadratne funkcije, koje imaju pozitivno definitnu Hesseovu matricu u svakoj točki, može se iterativno primjenjivati

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \eta \mathbf{H}_f(\mathbf{x}_i) \nabla_{\mathbf{x}} f(\mathbf{x}_i) \quad (2.84)$$

s $\eta < 1$.

3. Statističko zaključivanje

U ovom poglavlju su opisane neke od osnovnih ideja koje omogućuju efikasno **statističko zaključivanje**, tj. optimalno procjenjivanje nesigurnosti i svojstava neopaženih slučajnih varijabli na temelju pravila vjerojatnosti, opažanja i pretpostavki. Zbog nedostatka vremena, ovdje nije obrađeno statističko testiranje, koje je jedno važno područje statističkog zaključivanja.

Neka su x_1, \dots, x_n slučajne varijable čiju združenu razdiobu razmatramo. Želimo na temelju opeženih varijabli korištenjem pravila vjerojatnosti **zaključivati** o razdiobama nekih neopaženih varijabli. Općenito, zaključivanje se provodi uvjetovanjem po opaženim varijablama i marginalizacijom po varijablama koje nas ne zanimaju izravno (Murphy, 2012):

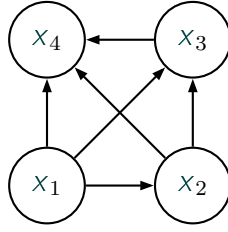
$$p(\mathbf{x}_q | \mathbf{x}_o) = \frac{p(\mathbf{x}_q, \mathbf{x}_o)}{p(\mathbf{x}_o)} = \frac{\int p(\mathbf{x}_q, \mathbf{x}_n, \mathbf{x}_o) d\mathbf{x}_n}{\int p(\mathbf{x}_q, \mathbf{x}_n, \mathbf{x}_o) d(\mathbf{x}_q, \mathbf{x}_n)}. \quad (3.1)$$

Ovdje je \mathbf{x}_q niz varijabli o kojima želimo zaključivati (varijable upita), \mathbf{x}_o niz opaženih varijabli, a \mathbf{x}_n niz varijabli *smetnje* (*nuisance*).

3.1. Probabilistički grafički modeli

Zavisnosti između slučajnih varijabli otežavaju modeliranje i zaključivanje — potrebno je više podataka i zaključivanje je računski zahtjevnije ako postoji puno zavisnosti. Obično možemo pretpostaviti uvjetne zavisnosti između slučajnih varijabli, što se može predstaviti neusmjerenim ili usmjerenim grafom. Prema definiciji na Wikipediji ¹, **probabilistički grafički model** ili **grafički model** je probabilistički model koji se može prikazati grafom koji izražava strukturu uvjetnih zavisnosti među slučajnim varijablama. U tom grafu čvorovi označavaju slučajne varijable, a bridovi izravne odnose između varijabli. Ako je graf grafičkog modela

¹https://en.wikipedia.org/wiki/Graphical_model



Slika 3.1: Prikaz grafičkog modela s faktorizacijom
 $p(x_1, x_2, x_3, x_4) = p(x_1) p(x_2 | x_1) p(x_3 | x_1, x_2) p(x_4 | x_1, x_2, x_3)$.

usmjeren i acikličan, on se naziva **Bayesova mreža** ili **Bayesovski model**, a ako je neusmjeren, naziva se **Markovljeva mreža** ili **Markovljevo slučajno polje** (engl. *Markov random field, MRF*). Neki odnosi uvjetnih (ne)zavisnosti mogu se modelirati jednom, a neki drugom vrstom grafičkih modela. U nastavku ovog odjeljka naglasak će biti na Bayesovim mrežama.

Združena razdioba se prema pravilu umnoška (jednadžba (2.6)) može npr. ovako izraziti:

$$p(x_1, \dots, x_n) = p(x_1) p(x_2 | x_1) \cdots p(x_n | x_1, \dots, x_{n-1}) \quad (3.2)$$

$$= \prod_i p(x_i | x_1, \dots, x_{i-1}). \quad (3.3)$$

Ako uzmemo $n = 4$, graf koji odgovara toj faktorizaciji je prikazan na slici 3.1.

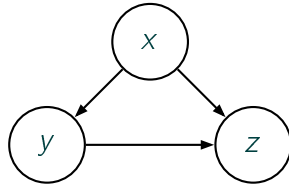
Pretpostavljanjem uvjetnih nezavisnosti, neki bridovi grafa G se mogu ukloniti, pa za varijable (čvorove grafa) vrijedi **uređajno Markovljevo svojstvo**, tj. slučajna varijabla je nezavisna o precima u grafu uz opažene roditelje. To se može ovako izraziti:

$$x \perp \text{pred}_G(x) \setminus \text{pa}_G(x) \mid \text{pa}_G(x), \quad (3.4)$$

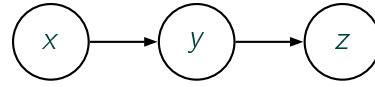
Gdje je $\text{pred}_G(x)$ skup predaka, a $\text{pa}_G(x)$ skup roditelja slučajne varijable x u grafu G . Jednadžba (3.3) onda prelazi u

$$p(x_1, \dots, x_n) = \prod_i p\left(x_i \mid \bigcap_{x_j \in \text{pa}_G(x_i)} \{x_j = x_j\}\right). \quad (3.5)$$

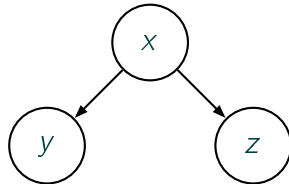
To omogućuje primjenu efikasnijih algoritama za zaključivanje (Murphy, 2012). Na slici 3.2 prikazani su osnovni slučajevi odnosa između triju slučajnih varijabli povezanih zavisnostima koje mogu biti dio većeg grafa. Oni su detaljnije objašnjeni npr. u Bishop (2006); Alpaydin (2014).



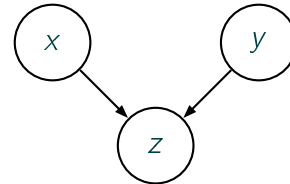
(a) Grafički model s faktorizacijom $p(x, y, z) = p(x) p(y | x) p(z | x, y)$.



(b) Uz $x \perp z | y$ faktorizacija postaje $p(x, y, z) = p(x) p(y | x) p(z | y)$ (lanac).



(c) Uz $y \perp z | x$ faktorizacija postaje $p(x, y, z) = p(x) p(y | x) p(z | x)$ (račvanje).



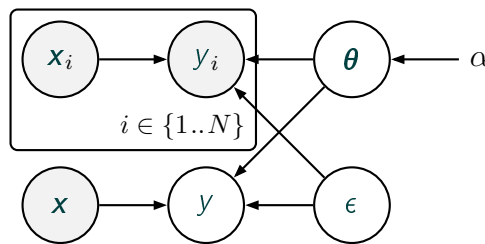
(d) Uz $x \perp y$ faktorizacija postaje $p(x, y, z) = p(x) p(y) p(z | x, y)$ (sraz). Ovdje također vrijedi $x \not\perp y | z$.

Slika 3.2: Osnovni slučajevi uvjetne nezavisnosti. Slike b, c i d prikazuju grafove dobivene uvođenjem pretpostavki uvjetne nezavisnosti za grafički model s 3 slučajne varijable prikazan na slici a.

Na slici 3.3 prikazan je primjer na kojem se koriste još neke oznake: sivi čvorovi označavaju opažene varijable, četverokut označava veći broj podgrafova s istom strukturom.

Općenitije, o uvjetnoj nezavisnosti podskupova varijabli govori svojstvo **d-separacije**. Kažemo da je staza (podgraf sa strukturom lanca) P grafa G **d-odvojena** skupom čvorova E akko P sadrži barem jedno od sljedećeg (Murphy, 2012):

1. lanac $a \rightarrow b \rightarrow c$, gdje $b \in E$



Slika 3.3: Primjer grafičkog modela s faktorizacijom $p(x, y, x_1 \dots x_N, y_1 \dots y_N, \theta, \epsilon) = p(\theta) p(\epsilon) p_x(x) p_{y|x, \theta, \epsilon}(y | x, \theta, \epsilon) \prod_i (p_{x_i}(x_i) p_{y_i|x, \theta, \epsilon}(y_i | x_i, \theta, \epsilon))$. Graf prikazuje model regresije, gdje su θ nepoznati parametri, x_i i y_i opaženi parovi ulaza i izlaza, x opaženi ulaz s nepoznatim izlazom y , a ϵ homoskedastički šum, tj. šum koji ne ovisi o ulazu. Na slici je još eksplicitno prikazana deterministička varijabla α koja je parametar razdiobe $p(\theta) = p(\theta | \alpha)$. Slika je napravljena po uzoru na sliku 14.7 u Alpaydin (2014).

2. račvanje $a \leftarrow b \rightarrow c$, gdje $b \in E$
3. sraz $a \rightarrow b \leftarrow c$, gdje $\forall b' \in \{b\} \cup \text{succ}_G(b), b' \notin E$.

Kažemo da skup čvorova E d-odvaja čvorove x i y akko su sve staze između njih d-odvojene. Vrijedi $x \perp y \mid E$ akko skup čvorova E d-odvaja čvorove x i y . To se može poopćiti na skupove čvorova. Skup čvorova opažanjem kojega neki čvor postaje neovisan o ostatku grafa naziva se **Markovljev pokrivač** (engl. *Markov blanket*). Markovljev pokrivač čvora x je

$$\text{pa}_G(x) \cup \text{ch}_G(x) \cup \bigcup_{y \in \text{ch}_G(x)} \text{pa}_G(y) \quad (3.6)$$

U knjigama navedenim u ovom odjeljku opisani su algoritmi koji se koriste za efikasno zaključivanje iskorištavanjem strukture grafa.

3.2. Procjena parametara i zaključivanje

3.2.1. Procjenitelji i točkaste procjene parametara

Ovaj pododjeljak se temelji na [Elezović \(2007\)](#).

Neka je x slučajna varijabla i $p(x)$ njena razdioba s nama nepoznatim parametrom θ . Taj parametar možemo procijeniti na temelju opaženih vrijednosti x_1, \dots, x_N slučajne varijable x , za što definiramo funkciju f koja daje procjenu parametara

$$\hat{\theta} = f(x_1, \dots, x_N). \quad (3.7)$$

Ako kao parametre takve funkcije uzmemo **uzorak**, tj. skup slučajnih varijabli $\mathcal{D} = (x_1, \dots, x_N)$, gdje pretpostavljamo da su x_1, \dots, x_N međusobno nezavisne i imaju istu razdiobu kao x , dobivamo slučajnu varijablu

$$\hat{\theta} = f(\mathcal{D}). \quad (3.8)$$

Takva slučajna varijabla naziva se **statistika**. Ako je θ nepoznati parametar razdiobe $p(x)$, onda kažemo da je ta statistika $\hat{\theta}$ **procjenitelj** parametra θ , a njen ishod $\hat{\theta}$ **procjena** parametra θ .

3.2.2. Svojstva i pogreška procjenitelja

Pristranost procjenitelja $\hat{\theta}$ je definirana izrazom $\mathbf{E} \hat{\theta} - \theta$, gdje je θ stvarna vrijednost parametra koji se procjenjuje. Ona mjeri koliko procjenitelj griješi neovisno o ishodu uzorka. Kažemo da je procjenitelj parametra θ **nepristran** ako je njegovo očekivanje jednako parametru koji procjenjuje:

$$\mathbf{E} \hat{\theta} = \theta. \quad (3.9)$$

Varijanca procjenitelja $\hat{\theta}$ je definirana izrazom $\mathbf{D} \hat{\theta}$. Ona mjeri koliko procjenitelj griješi ovisno o variranju uzorka. Neka N u oznaci \mathcal{D}_N označava veličinu uzorka. Nepistrani procjenitelj $\hat{\theta}$ je **valjan** ako

$$\lim_{N \rightarrow \infty} \mathbf{D}[\hat{\theta}(\mathcal{D}_N)] = 0. \quad (3.10)$$

Očekivanje srednje kvadratne pogreške procjenitelja je jednako zbroju njegove varijance i kvadrata njegove pristranosti tj.

$$\mathbf{E}((\hat{\theta} - \theta)^2) = \mathbf{D} \hat{\theta} + (\mathbf{E} \hat{\theta} - \theta)^2, \quad (3.11)$$

što možemo pokazati uz korištenje linearnosti očekivanja i izražavanja varijance prema jednadžbi (2.22):

$$\mathbf{E}((\hat{\theta} - \theta)^2) = \mathbf{E}(\hat{\theta}^2 - 2\theta\hat{\theta} + \theta^2) \quad (3.12)$$

$$= \mathbf{E} \hat{\theta}^2 - 2\theta \mathbf{E} \hat{\theta} + \theta^2 \quad (3.13)$$

$$= \underbrace{\mathbf{E} \hat{\theta}^2 - (\mathbf{E} \hat{\theta})^2}_{\mathbf{D} \hat{\theta}} + \underbrace{(\mathbf{E} \hat{\theta})^2 - 2\theta \mathbf{E} \hat{\theta} + \theta^2}_{(\mathbf{E} \hat{\theta} - \theta)^2}. \quad (3.14)$$

3.2.3. Procjenitelj maksimalne izglednosti

Procjenitelj maksimalne izglednosti (ML-procjenitelj, engl. *maximum likelihood*) uzorku dodjeljuje parametre maksimiziraju vjerojatnost uzorka, tj. imaju najveću **izglednost**:

$$\theta_{\text{ML}} = \arg \max_{\theta} p(\mathcal{D} \mid \theta). \quad (3.15)$$

Zbog pretpostavke međusobne nezavisnosti primjera vrijedi

$$p(\mathcal{D} \mid \theta) = \prod_{d \in \mathcal{D}} p(d \mid \theta). \quad (3.16)$$

Za razliku od generativnih, diskriminativni modeli ne modeliraju razdiobu ulaznih primjera, nego samo uvjetnu razdiobu $p(\mathbf{y} \mid \mathbf{x}, \mathcal{D})$, pa kod njih razdioba ulaznih primjera ne ovisi o θ , tj. $p(\mathbf{x} \mid \theta) = p(\mathbf{x})$. Onda je izglednost

$$p(\mathcal{D} \mid \theta) = \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} p(\mathbf{y} \mid \mathbf{x}, \theta) p(\mathbf{x} \mid \theta) = p(\mathbf{x}) \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} p(\mathbf{y} \mid \mathbf{x}, \theta). \quad (3.17)$$

Faktor $p(\mathbf{x})$ ne ovisi o parametrima i može se zanemariti pri optimizaciji.

3.2.4. Procjenitelj maksimalne aposteriorne vjerojatnosti

Procjenitelj maksimalne aposteriorne vjerojatnosti (MAP-procjenitelj, engl. *maximum a posteriori estimator*) u obzir uzima **apriornu razdiobu** $p(\theta)$ koja predstavlja dodatne pretpostavke za razdiobu parametara. Apriorna razdioba parametara pojednostavljuje model dajući prednost nekim hipotezama i posebno je korisna kada ima malo podataka. Apriorna razdioba može biti definirana nekim hiperparametrima ali oni ovdje nisu prikazani. Po Bayesovom pravilu, **aposteriorna vjerojatnost** parametara je

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{\int p(\mathcal{D} \mid \theta') p(\theta') d\theta'}. \quad (3.18)$$

Maksimizacijom aposteriorne vjerojatnosti dobivaju se parametri

$$\theta_{\text{MAP}} = \arg \max_{\theta} p(\theta \mid \mathcal{D}) = \arg \max_{\theta} p(\mathcal{D} \mid \theta) p(\theta). \quad (3.19)$$

Ovdje nije potrebno normalizirati aposteriornu vjerojatnost izračunavanjem **marginalne izglednosti** (engl. *marginal likelihood, evidence*) $p(\mathcal{D})$ u nazivniku na desnoj strani jednadžbe (3.18) jer ona ne ovisi o θ , nego samo o modelu \mathcal{H} . Odabirom uniformne (neinformativne) apriorne razdiobe MAP-procjenitelj postaje ekvivalentan ML-procjenitelju.

Poželjno je da $p(\mathcal{D} \mid \theta)$ i $p(\theta)$ kao funkcije parametra θ imaju takav algebarski oblik da njihov umnožak ima sličan oblik i može se analitički izračunati. Ako $p(\theta)$ i $p(\theta \mid \mathcal{D})$ imaju isti algebarski oblik, nazivaju se **konjugatne razdiobe** (Šnajder i Dalbelo Bašić, 2014).

3.2.5. Bayesovski procjenitelj i zaključivanje

Prethodno opisani procjenitelji daju točkastu procjenu parametara i ne izražavaju nesigurnost procjene kojoj uzrok može biti npr. nedovoljna količina podataka ili šum u podacima za učenje. **Bayesovski procjenitelj** prema jednadžbi (3.18) kao procjenu daje razdiobu nad hipotezama $p(\boldsymbol{\theta} \mid \mathcal{D})$ za koju je integriranjem po svim mogućim parametrima potrebno izračunati marginalnu izglednost $p(\mathcal{D}) = \int p(\mathcal{D} \mid \boldsymbol{\theta}') p(\boldsymbol{\theta}') d\boldsymbol{\theta}'$ u nazivniku.

Kod složenijih modela često ne možemo odabrati konjugatnu apriornu razdiobu, a i funkcija izglednosti je sama po sebi već dovoljno složena da se, neovisno o apriornoj razdiobi, marginalna izglednost $p(\mathcal{D})$ ne može ni analitički ni numerički traktabilno računati.

Vjerojatnost nekog primjera \mathbf{d} procjenjuje se marginalizacijom po parametrima (Neal, 1995):

$$p(\mathbf{d} \mid \mathcal{D}) = \int p(\mathbf{d} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} = \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} p(\mathbf{d} \mid \boldsymbol{\theta}), \quad (3.20)$$

gdje je korištena uvjetna nezavisnost $\mathbf{d} \perp \mathcal{D} \mid \boldsymbol{\theta}$ (zbog čega $p(\mathbf{d} \mid \boldsymbol{\theta}, \mathcal{D}) = p(\mathbf{d} \mid \boldsymbol{\theta})$).

Kada se parametri točkasto procjenjuju, npr. MAP-procjeniteljem, točkasta procjena parametara $\hat{\boldsymbol{\theta}}$ aproksimira cijelu aposteriornu razdiobu, tj. $p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$. Onda je

$$p(\mathbf{d} \mid \mathcal{D}) \approx \int p(\mathbf{d} \mid \boldsymbol{\theta}) \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) d\boldsymbol{\theta} = p(\mathbf{d} \mid \hat{\boldsymbol{\theta}}). \quad (3.21)$$

Za diskriminativne modele se zaključivanje može izraziti ovako:

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} = \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}), \quad (3.22)$$

gdje je korištena uvjetna nezavisnost $\mathbf{y} \perp \mathcal{D} \mid \boldsymbol{\theta}$ — ako su nam poznati parametri ili njihova razdioba dobivena na temelju uzorka, uvjetna razdioba $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D})$ ne ovisi o uzorku \mathcal{D} .

Kod regresije je često, ako pretpostavljamo da pogreška izlaza ima Gaussovu razdiobu, najbolja procjena hipoteze očekivanje po naučenoj razdiobi parametara:

$$h(\mathbf{x}) = \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} h(\mathbf{x}; \boldsymbol{\theta}) = \int h(\mathbf{x}; \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta}, \quad (3.23)$$

koje je ujedno i MAP-hipoteza. U tom slučaju se nesigurnost može izraziti

varijancom $\mathbf{D}_{\theta|\mathbb{D}} h(\mathbf{x}; \theta)$.

3.3. Monte Carlo aproksimacija

Ovaj odjeljak se temelji na [Goodfellow et al. \(2016\)](#), pododjeljak 17.1.2).

Monte Carlo aproksimacija je postupak procjenjivanja vrijednosti koje se mogu izraziti kao očekivanje neke funkcije neke slučajne varijable na temelju uzoraka. Ponekad nije moguće analitički ili numerički traktabilno ili efikasno izračunati neki integral (ili zbroj). Ako taj integral ili zbroj može ovako izraziti:

$$s = \int p(x)f(x) dx = \mathbf{E} f(x), \quad (3.24)$$

možemo ga procijeniti uzorkovanjem:

$$\hat{s}_n = \frac{1}{n} \sum_{i=1}^n f(x_i). \quad (3.25)$$

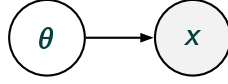
Procjenitelj \hat{s}_n je nepristran ako su x_i nezavisne i imaju istu razdiobu kao x . On je i valjan ako su varijance $f(x_i)$ ograničene. Ako su sve varijance jednake, vrijedi $\mathbf{D} \hat{s}_n = \frac{1}{n} \mathbf{D} f(x)$.

U širem smislu, postupci *Monte Carlo* obuhvaćaju i generiranje uzoraka slučajne varijable čije se očekivanje procjenjuje.

3.4. Aproksimacija razdioba i aproksimacijsko zaključivanje

Ovaj odjeljak se uglavnom temelji na [Blei et al. \(2017\)](#) i malo na [Yang \(2017\)](#).

Važan problem u bayesovskoj statistici, gdje se zaključivanje temelji na izračunima koji uključuju aposteriornu razdiobu, je aproksimacija razdioba koje su zahtjevnije za računanje. Kod složenijih Bayesovskih modela aposteriorna razdioba se ne može lako izračunati i treba koristiti aproksimacijske postupke od kojih su glavni **varijacijski** postupci ([Jordan et al., 1999](#)) i postupci **Monte Carlo** aproksimacije s uzorkovanjem pomoću **Markovljevog lanca** (MCMC, engl. *Markov chain Monte Carlo*). MCMC-postupci temelje se na definiranju stohastičkog procesa koji ima stacionarnu razdiobu jednaku razdiobi koja se aproksimira, omogućuju asimptotski



Slika 3.4: Prikaz grafičkog modela sa skrivenom varijablom θ i opaženom varijablom x .

egzaktno uzorkovanje velike klase razdioba. Varijacijski postupci temelje se na aproksimaciji razdiobe nekom jednostavnijom koja se pronalazi rješavanjem optimizacijskog problema, brži su i jednostavniji za ostvariti za složenije modele.

Razmatramo bayesovski model koji ima latentnu varijablu θ i vidljivu varijablu x . Model je prikazan na slici 3.4 i opisan je ovom jednadžbom združene vjerojatnosti:

$$p(\mathbf{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) p(\mathbf{x} | \boldsymbol{\theta}).$$

Zaključivanjem se određuje aposteriorna razdioba latentne varijable:

$$p(\boldsymbol{\theta} | \mathbf{x}) = \frac{p(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{x})} = \frac{p(\mathbf{x}, \boldsymbol{\theta})}{\int p(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}}. \quad (3.26)$$

na temelju opaženih vrijednosti slučajne varijable x (podataka). Kod složenijih modela integriranje marginalne izglednosti u nazivniku nije traktabilno i aposteriorna razdioba se mora aproksimirati **približnim (aproksimacijskim) zaključivanjem**.

3.5. Varijacijsko zaključivanje

Za razliku od uzorkovanja kod MCMC-postupaka, osnovna ideja kod varijacijskog zaključivanja je optimizacija. Prvo se odabire familija razdioba $\mathcal{Q} = \{q_\phi\}_\phi$ koje su lakše za računanje. Razdiobe iz \mathcal{Q} su parametrizirane tzv. **varijacijskim parametrima** ϕ . Cilj je na temelju podataka kao zamjenu za aposteriornu razdiobu $p(\boldsymbol{\theta} | \mathbf{x})$ pronaći razdiobu iz \mathcal{Q} koja ju što bolje aproksimira. To možemo ostvariti minimizacijom Kullback-Leiblerove (KL) divergenciju s obzirom na stvarnu aposteriornu razdiobu po varijacijskim parametrima ϕ :

$$q^* = \arg \min_{q_\phi} D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta} | \mathbf{x})), \quad (3.27)$$

Naziv **varijacijsko zaključivanje** dolazi od varijacijskog računa², gdje se koriste varijacije, tj. male promjene u funkcijama i funkcionalima³, kako bi se pronašli

²https://en.wikipedia.org/wiki/Calculus_of_variations

³Funkcionalni su preslikavanja iz skupa funkcija u \mathbb{R} . Oni su često izraženi kao integrali koji uključuju funkcije i njihove derivacije.

minimumi ili maksimumi funkcionala.

Ako ciljnu funkciju ovako izrazimo:

$$\begin{aligned} D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta} \mid \mathbf{x})) &= \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln \frac{q_\phi(\tilde{\boldsymbol{\theta}})}{p_{\boldsymbol{\theta} \mid \mathbf{x}}(\tilde{\boldsymbol{\theta}})} \\ &= \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln q_\phi(\tilde{\boldsymbol{\theta}}) - \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln p(\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}, \mathbf{x}) + \ln p(\mathbf{x}), \end{aligned} \quad (3.28)$$

vidi se da se ona ne može lako izračunati jer zahtijeva računanje marginalne izglednosti $p(\mathbf{x})$ iz nazivnika u jednadžbi (3.26) marginalizacijom po $\boldsymbol{\theta}$. Marginalna izglednost ne ovisi o varijacijskim parametrima, pa ju možemo zanemariti i maksimiziramo funkciju koja se naziva **varijacijska donja granica** (engl. *variational lower bound*) ili **donja granica (logaritma) marginalne izglednosti** (engl. *(log) evidence lower bound, ELBO*) (Jordan et al., 1999):

$$L_x(q_\phi) := \ln p(\mathbf{x}) - D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta} \mid \mathbf{x})) \quad (3.29)$$

$$= \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln p(\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}, \mathbf{x}) - \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln q_\phi(\tilde{\boldsymbol{\theta}}). \quad (3.30)$$

Naziv *donja granica logaritma marginalne izglednosti* dolazi od toga što vrijedi $L_x(q_\phi) \leq \ln p(\mathbf{x})$. To slijedi iz jednadžbe (3.29) i nenegativnosti KL-divergencije:

$$\ln p(\mathbf{x}) = L_x(q_\phi) + D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta} \mid \mathbf{x})) \geq L_x(q_\phi). \quad (3.31)$$

Donja granica marginalne izglednosti se može i ovako izraziti:

$$L_x(q_\phi) = \mathbb{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} \ln p(\mathbf{x} \mid \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) - D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta})). \quad (3.32)$$

Maksimiziranje takve ciljne funkcije s obzirom na varijacijske parametre daje razdiobu koja dobro objašnjava podatke jer se potiče veće očekivanje logaritma izglednosti (prvi član), a ne razlikuje se previše od apriorne razdiobe jer se potiče manja KL-divergencija između varijacijske razdiobe i apriorne razdiobe (Gal i Ghahramani, 2015b).

3.5.1. Metoda srednjeg polja

Dodatno pojednostavljenje koje pomaže u modeliranju i optimizaciji je pretpostavljanje nezavisnosti između latentnih varijabli. Onda za varijacijsku

razdiobu vrijedi ovakva faktorizacija:

$$q_{\phi}(\tilde{\theta}) = \prod_i q_{\phi,i}(\tilde{\theta}_{[i]}), \quad (3.33)$$

gdje su $q_{\phi,i}$ funkcije gustoće pojedinih slučajnih varijabli. Kod **metode srednjeg polja** pretpostavlja se takva razdioba i obično se primjenjuje koordinatni spust za optimizaciju. To je detaljnije objašnjeno npr. u [Murphy \(2012\)](#).

4. Nadzirano strojno učenje

Ovo poglavlje se uglavnom temelji na Šnajder i Dalbelo Bašić (2014); Goodfellow et al. (2016).

Zadatak algoritama **nadziranog učenja** je preslikavanje **ulaznih primjera** x iz **ulaznog prostora** \mathbb{X} u **izlaze (oznake)** $y \in \mathbb{Y}$ na temelju konačnog skupa označenih primjera $\mathcal{D} = \{(x_i, y_i)\}_i$. Algoritmima strojnog učenja pretražuje se **model** ili **prostor hipoteza** u cilju pronalaska **hipoteze** koja osim primjera iz skupa za učenje, u izlaze dobro preslikava i primjere koji nisu u skupu za učenje. Sposobnost postizanja dobre performanse na neviđenim primjerima naziva se **generalizacija**.

Neka je $\mathcal{D} = \{d_i\}_i$ skup nezavisnih primjera izvučenih iz neke razdiobe \mathcal{D} . Možemo definirati **probabilistički model** \mathcal{H} s nepoznatim parametrima θ kojem je cilj što bolje modelirati tu razdiobu pronalaskom najbolje hipoteze na temelju podataka: $p(d \mid \mathcal{D}, \mathcal{H})$. Model koji modelira razdiobu primjera nazivamo **generativnim modelom**. U nastavku ćemo izostavljati oznaku modela radi kraćeg zapisa.

Ako su primjeri parovi $d_i = (x_i, y_i) \in \mathbb{X} \times \mathbb{Y}$, može nam biti cilj ulaznim primjerima iz \mathbb{X} dodjeljivati oznake iz \mathbb{Y} . Ako je problem koji rješavamo dodjeljivanje oznaka ulaznim primjerima, onda su često prikladniji **diskriminativni modeli**. Probabilistički diskriminativni modeli izravno modeliraju uvjetne razdiobe $p(y \mid x)$ hipotezom koja ulazni primjer x preslikava u razdiobu $p(y \mid x, \mathcal{D})$. Neprobabilistički diskriminativni modeli modeliraju funkciju dodjeljivanja oznaka hipotezom $h: \mathbb{X} \rightarrow \mathbb{Y}$. Modeliranje zajedničke razdiobe $p(x, y)$ obično zahtijeva više računalnih resursa i podataka (Bishop, 2006).

Modeli se još mogu podijeliti na **parametarske** i **neparametarske**. Kod parametarskih modela broj parametara je unaprijed određen, dok kod neparametarskih on ovisi o podacima za učenje.

4.1. Induktivna pristranost

Uz zadani skup hipoteza koji dopušta model, **algoritam strojnog učenja** traži parametre koji definiraju jednu hipotezu. Učenje hipoteze je loše definiran (engl. *ill-posed*) problem jer skup podataka \mathcal{D} nije dovoljan za jednoznačan odabir hipoteze. Osim dobrog opisivanja podataka za učenje, naučena hipoteza mora dobro generalizirati. Kako bi učenje i generalizacija bili mogući, potreban je skup pretpostavki koji se naziva **induktivna pristranost**. Razlikujemo dvije vrste induktivne pristranosti (Šnajder i Dalbelo Bašić, 2014):

1. **pristranost ograničavanjem** ili **pristranost jezika** — ograničavanje skupa hipoteza koje se mogu prikazati modelom
2. **pristranost preferencijom** ili **pristranost pretraživanja** — dodjeljivanje različitih prednosti različitim hipotezama.

Većina algoritama strojnog učenja kombinira obje vrste induktivne pristranosti.

4.2. Komponente algoritma strojnog učenja

Prema Šnajder i Dalbelo Bašić (2014), kod većine algoritama strojnog učenja možemo razlikovati 3 osnovne komponente, od kojih prva predstavlja pristranost ograničavanjem, a druge dvije obično pristranost preferencijom:

1. **Model** ili prostor hipoteza. Model \mathcal{H} je skup funkcija h parametriziranih parametrima θ : $\mathcal{H} = \{h(\mathbf{x}; \theta)\}_{\theta}$.
2. **Funkcija pogreške** ili ciljna funkcija. Funkcija pogreške $E(\theta, \mathcal{D})$ na temelju parametara modela (hipoteze) i skupa podataka izračunava broj koji izražava procjenu dobrote hipoteze. Obično pretpostavljamo da su primjeri iz skupa za učenje nezavisni i definiramo **funkciju gubitka** $L: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, kojoj je prvi parametar predikcija hipoteze, a drugi ciljna oznaka koja odgovara ulaznom primjeru. Funkciju pogreške možemo definirati kao prosječni gubitak na skupu za učenje:

$$E(\theta, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} L(\mathbf{y}, h(\mathbf{x}; \theta)). \quad (4.1)$$

Obično joj dodajemo **regularizacijski** član kojim unosimo dodatne pretpostavke radi postizanja bolje generalizacije. Više o funkciji pogreške u

smislu smanjivanja empirijskog i strukturnog rizika piše u odjeljku 4.4.

3. **Optimizacijski postupak.** Optimizacijski postupak je algoritam kojim pronalazimo hipotezu koja minimizira pogrešku:

$$\theta^* = \arg \min_{\theta} E(\theta, \mathcal{D}). \quad (4.2)$$

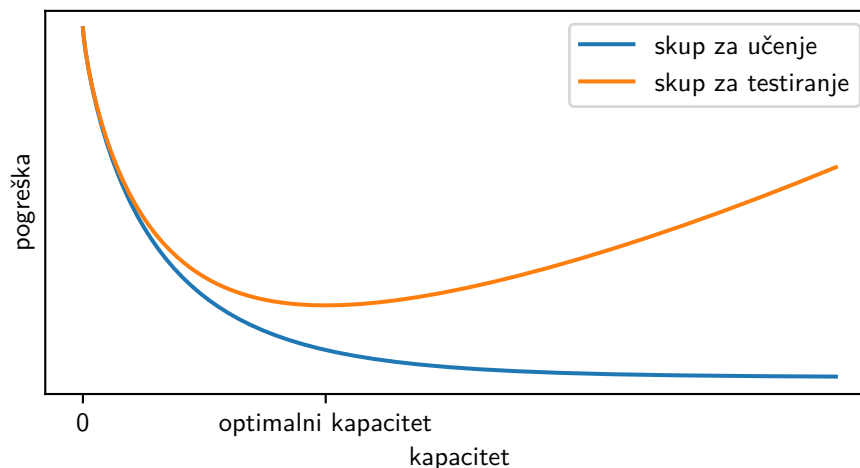
Kod nekih jednostavnijih modela minimum možemo odrediti analitički. Inače moramo koristiti neki iterativni optimizacijski postupak. Kod nekih složenijih modela, kao što su neuronske mreže, funkcija pogreške nije unimodalna i vjerojatnost pronalaska globalnog optimuma je zanemariva, ali ipak se mogu pronaći dobra rješenja.

U literaturi riječ *model* često ima šire značenje. Uz skup hipoteza obično obuhvaća i induktivnu pristranost ili dio nje. Model u tom smislu bi se formalno mogao definirati kao par (\mathcal{H}, B) , gdje je \mathcal{H} skup mogućih hipoteza, a B induktivna pristranost koja hipotezama dodjeljuje različite važnosti. Npr. za regresiju \mathcal{H} može biti skup polinoma stupnja 4, a B pretpostavka da primjeri imaju razdiobu $p(y | x) = \mathcal{N}(h(x), 1)$, gdje je h nepoznata hipoteza. Ovdje će se u nastavku koristiti takvo značenje riječi *model*, a riječ *prostor hipoteza* će se koristiti sa značenjem modela u užem smislu.

4.3. Kapacitet modela, podnaučenost i prenaučenost

Cilj algoritama strojnog učenja je postići malu **pogrešku generalizacije**, tj. malo očekivanje pogreške na primjera koji nisu korišteni za učenje i odabir modela. Generalizacijska pogreška se procjenjuje pogreškom na skupu podataka koji nije korišten za učenje. Obično pretpostavljamo da su skupovi primjera koje koristimo za učenje, odabir modela i testiranje generirani međusobno nezavisno i iz iste razdiobe.

Kapacitet ili složenost modela je svojstvo koje opisuje njegovu sposobnost prilagodbe podacima. Model koji se previše prilagođava podacima za učenje (i statističkom šumu u njima) obično ima slabu prediktivnu moć. Treba odabrati model (ili hipotezu) koji dobro objašnjava podatke, ali nije previše složen. O tome govori i načelo **Occamove oštrice** prema kojem među hipotezama konzistentnima s opažanjem treba odbaciti sve osim najjednostavnije od njih. Postoje formalizacije



Slika 4.1: Ovisnost pogrešaka na skupovima za učenje i testiranje o kapacitetu modela. Povećavanjem kapaciteta povećava se razlika između pogreške na skupovima za testiranje i pogreške na skupovima za učenje.

Occamove oštrice (Blumer et al., 1987, 1989; Grünwald, 2005; Rathmanner i Hutter, 2011). Na ograničavanje složenosti modela možemo utjecati ograničavanjem prostora hipoteza i regularizacijom (*mekim* ograničavanjem).

Model s većim kapacitetom (složeniji model) može postići manju pogrešku na skupovima za učenje. Prevelik kapacitet povećava pogrešku generalizacije. Za model koji daje veliku pogrešku generalizacije kažemo da je **prenaučen**. Kod takvog modela hipoteze će jako varirati u ovisnosti o skupovima za učenje i zato kažemo da složeni modeli imaju visoku varijancu. Model premalog kapaciteta (prejednostavan model) ima manju razliku između pogreške na skupovima za učenje i pogreške na skupovima za testiranje, ali su obje pogreške veće od optimalnih. Za model koji ne postiže malu pogrešku na skupovima za učenje kažemo da je **podnaučen**. U jednostavan model ugrađene su jače pretpostavke i kažemo da on ima jaču pristranost. Uobičajena ovisnost pogrešaka na skupovima za učenje i testiranje o kapacitetu ilustrirana je slikom 4.1.

4.4. Rizik i funkcija pogreške

Dijelovi ovog odjeljka temelje se na Murphy (2012, podjeljak 6.5).

4.4.1. Rizik i empirijski rizik

Zadatak nadziranog strojnog učenja može se formulirati kao optimizacijski problem minimizacije **rizika**. Neka su θ odabrani parametri. Definiramo **funkciju gubitka** $L: \mathbb{Y} \times \mathbb{Y} \rightarrow \mathbb{R}$ koja kažnjava neslaganje izlaza sa stvarnom oznakom. **Rizik** definiramo kao očekivanje funkcije gubitka:

$$R(\theta; \mathcal{D}) = \mathbf{E}_{(x,y) \sim \mathcal{D}} L(y, h(x; \theta)). \quad (4.3)$$

Razdioba koja generira podatke nije poznata, pa se koristi **empirijski rizik** koji **prirodnu razdiobu** \mathcal{D} procjenjuje **empirijskom razdiobom**, tj. uzorkom \mathbb{D} :

$$R_E(\theta; \mathbb{D}) = \mathbf{E}_{(x,y) \sim \mathbb{D}} L(y, h(x; \theta)) = \frac{1}{|\mathbb{D}|} \sum_{(x,y) \in \mathbb{D}} L(y, h(x; \theta)). \quad (4.4)$$

Što je uzorak veći \mathbb{D} , sličniji je prirodnoj razdiobi i procjena rizika je bolja. U slučaju nenadziranog učenja, kada se hipoteza sastoji od kodera E i dekodera D , tj. $h(x; \theta) = E(D(x; \theta); \theta)$, ili generativnog modela, kada je $h(x; \theta) = p(x | \theta)$, gubitak mjeri **pogrešku rekonstrukcije** i izraz za rizik je (Murphy, 2012):

$$R(\theta; \mathcal{D}) = \mathbf{E}_{d \sim \mathcal{D}} L(d, h(d; \theta)). \quad (4.5)$$

Kod probabilističkih modela empirijski rizik se može definirati kao **negativni logaritam izglednosti** parametara:

$$R_E(\theta; \mathbb{D}) = -\frac{1}{|\mathbb{D}|} \ln p(\mathbb{D} | \theta) = -\frac{1}{|\mathbb{D}|} \sum_{d \in \mathbb{D}} \ln p(d | \theta), \quad (4.6)$$

gdje je korištena pretpostavka međusobne nezavisnosti primjera. Gubitak je onda $L(d, h(d; \theta)) = -\ln p(d | \theta)$. U slučaju diskriminativnog modela, uz zanemarivanja faktora izglednosti koji ne ovisi o θ (jednadžba (3.17)), vrijedi $L(d, h(x; \theta)) = -\ln p(y | x, \theta)$. Minimizacija gubitka definiranog kao negativni logaritam izglednosti ekvivalentna je minimizaciji KL-divergencije ili unakrsne entropije (odjeljak 2.2) s obzirom na empirijsku razdiobu. Zbog toga se takav gubitak još naziva **gubitak unakrsne entropije**.

4.4.2. Strukturni rizik i regularizacija

Kada ima malo podataka ili je model previše složen, minimizacija empirijskog rizika dovodi do velike varijance i slabe generalizacije. Procjenitelj koji minimizira

empirijski rizik ne uzima u obzir apriornu razdiobu parametara. Radi postizanja bolje generalizacije, funkciji pogreške dodaje se **regularizacijski gubitak** $\lambda R_R(\boldsymbol{\theta})$, $\lambda \geq 0$, koji predstavlja **strukturni rizik** koji daje prednost jednostavnijim hipotezama. Funkcija pogreške onda ima ovakav oblik:

$$E(\boldsymbol{\theta}; \mathcal{D}) = R_E(\boldsymbol{\theta}; \mathcal{D}) + \lambda R_R(\boldsymbol{\theta}). \quad (4.7)$$

Regularizacijski gubitak obično ovisi samo o parametrima, ali može ovisiti i o podacima (Goodfellow et al., 2016).

Ako kao funkciju pogreške koristimo negativni logaritam aposteriorne vjerojatnosti uz pretpostavku međusobne nezavisnosti primjera, funkcija pogreške je

$$E(\boldsymbol{\theta}; \mathcal{D}) = -\frac{1}{|\mathcal{D}|} \ln p(\boldsymbol{\theta} | \mathcal{D}) \quad (4.8)$$

$$= \underbrace{-\frac{1}{|\mathcal{D}|} \ln p(\mathcal{D} | \boldsymbol{\theta})}_{R_E(\boldsymbol{\theta}; \mathcal{D})} - \underbrace{\frac{1}{|\mathcal{D}|} \ln p(\boldsymbol{\theta})}_{\lambda R_R(\boldsymbol{\theta})} + C_1, \quad (4.9)$$

gdje je $C_1 = \frac{1}{|\mathcal{D}|} \ln p(\mathcal{D})$ konstanta koja ne ovisi o $\boldsymbol{\theta}$. Hiperparametar λ je onda parametar apriorne razdiobe. Možemo ga ovako izlučiti:

$$\ln p(\boldsymbol{\theta}) =: \lambda \ln p_0(\boldsymbol{\theta}) + C_2 = \ln p_0(\boldsymbol{\theta})^\lambda + C_2, \quad (4.10)$$

gdje je p_0 neka razdioba (funkcija gustoće), a $C_2 = -\ln(\int_{\boldsymbol{\theta}'} p_0(\boldsymbol{\theta}') d\boldsymbol{\theta}')$ konstanta koja ne ovisi o $\boldsymbol{\theta}$. Vidi se da λ određuje koncentraciju apriorne razdiobe. Povećanje λ smanjuje entropiju apriorne razdiobe. Ona postaje koncentriranija i regularizacija jača. Jačom regularizacijom se povećava pristranost i smanjuje varijanca.

Optimalne hiperparametre tražimo postupcima odabira modela (odjeljak 4.5) kod kojih se za procjenu generalizacije koriste podaci koji nisu korišteni za učenje.

4.5. Odabir modela

Ovaj odjeljak se temelji na Šnajder i Dalbelo Bašić (2014, odjeljak 2.6).

Performansa modela se mjeri nekom evaluacijskom mjerom. Ona omogućuje usporedbu hipoteza ili modela na nekom skupu podataka. Budući da nas zanima generalizacija, za procjenu generalizacije trebamo koristiti podatke koji nisu korišteni za učenje. Odabir modela se obično svodi na traženje optimalnih **hiperparametara**.

4.5.1. Unakrsna validacija

Najjednostavniji način procjenjivanja generalizacije je **unakrsna validacija**. Kod unakrsne validacije, skup podatakama dijelima na **skup za učenje** i **skup za validaciju**. Ako se unakrsna validacija ne koristi za odabir modela, nego za konačnu procjenu generalizacije, onda se skup na kojem se model evaluira naziva **skup za testiranje**.

Za dobivanje bolje procjene generalizacije često se koristi K -struka unakrsna validacija. Kod K -**struke unakrsne validacije** skup podataka \mathcal{D} se podijeli na K dijelova \mathcal{D}_i , $i = 1..K$. Model se uči K puta tako da se u i -toj iteraciji za skup za validaciju odabere \mathcal{D}_i , a za skup za učenje ostali podaci, $\mathcal{D} \setminus \mathcal{D}_i$. Kao konačna procjena generalizacije uzima se prosjek evaluacija iz svih iteracija.

4.5.2. Bayesovska usporedba modela

Ovaj pododjeljak se temelji na [Murray i Ghahramani \(2005\)](#) i ([Murphy, 2012](#), odjeljak 5.3).

Ako u izraz s desne strane jednadžbe 3.18 eksplicitno uključimo ovisnost parametara o modelu, imamo ovo:

$$p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{H}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}, \mathcal{H}) p(\boldsymbol{\theta} \mid \mathcal{H})}{p(\mathcal{D}, \mathcal{H})}, \quad (4.11)$$

gdje je \mathcal{D} skup podataka, \mathcal{H} model, a $\boldsymbol{\theta}$ parametri. Kao za parametre modela, možemo imati i aposteriornu razdiobu modela:

$$p(\mathcal{H} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{H}) p(\mathcal{H})}{p(\mathcal{D})}, \quad (4.12)$$

gdje je $p(\mathcal{D})$ konstanta neovisna o modelu. Ako pretpostavimo neinformativnu apriornu razdiobu $p(\mathcal{H})$, modele možemo uspoređivati na temelju marginalnih izglednosti integriranjem po svim mogućim parametrima:

$p(\mathcal{D} \mid \mathcal{H}) = \int_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}, \mathcal{H}) p(\boldsymbol{\theta} \mid \mathcal{H}) d\boldsymbol{\theta}$. Složeniji modeli vjerojatnost raspoređuju po većem broju skupova podataka i, ako je model presložen, marginalna izglednost na promatranom skupu će biti mala. To se naziva **bayesovska Occamova oštrica** ([MacKay, 1992a](#)).

4.6. Osnovni zadaci nadziranog učenja

Osnovni zadaci nadziranog učenja su **klasifikacija** i **regresija**. Zadatak klasifikacije je svakom ulaznom primjerima dodjeljivati oznake iz konačnog skupa oznaka, npr. $\{1..C\}$, gdje svaka oznaka predstavlja jednu **klasu (razred)**. Zadatak regresije je ulaznim primjerima dodjeljivati vrijednosti iz kontinuiranog skupa (obično \mathbb{R} ili \mathbb{R}^n). Ulazni primjeri su obično realni vektori. Klasifikacijska hipoteza se može definirati preko funkcije s kontinuiranom kodomenom. Ako $C = 2$, ta funkcija može biti $h: \mathbb{X} \rightarrow \mathbb{R}$, a hipoteza $h_c(\mathbf{x}) = \mathbb{I}[h(\mathbf{x}) > 0]$. Ako $C > 2$, onda to može biti npr. $h_c(\mathbf{x}) = \arg \max_i h_i(\mathbf{x})$, gdje $h: \mathbb{X} \rightarrow \mathbb{R}^C$ i $h(\mathbf{x}) = [h_i(\mathbf{x})]_{i=1..C}^T$. Kod nekih zadataka ulazi ili izlazi imaju složeniju strukturu i ona se može razlikovati između različitih primjera.

4.7. Primjeri modela: poopćeni linearni modeli

Ovaj odjeljak se temelji na Šnajder i Dalbelo Bašić (2014, odjeljak 6.1) i Šnajder (2017).

Linearni modeli su modeli kod kojih je hipoteza definirana afinom transformacijom:

$$h(\mathbf{x}) = h(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{w}^T \mathbf{x} + b, \quad (4.13)$$

gdje je \mathbf{w} vektor **težina**, b **pomak** (engl *bias*), a $\boldsymbol{\theta} = (\mathbf{w}, b)$. Kod linearnih modela je, u slučaju klasifikacije, granica $(n - 1)$ -dimenzionalna hiperravnina s normalom \mathbf{w} . Obično se na ulazne primjere primjenjuje neka nelinearna transformacija

$$\begin{aligned} \phi: \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ \mathbf{x} &\mapsto [\phi_1(\mathbf{x}), \dots, \phi_m(\mathbf{x})]^T \end{aligned}$$

koja predstavlja preslikavanje ulaznog prostora u **prostor značajki**. Funkcije $\phi_i: \mathbb{R}^n \rightarrow \mathbb{R}$ nazivaju se **bazne funkcije**. Hipoteza linearnog modela onda ima oblik

$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}). \quad (4.14)$$

Ovdje je izostavljen pomak b jer jedan od izlaza transformacije ϕ može biti konstanta 1 koja se množi s jednom težinom iz \mathbf{w} .

Poopćeni linearni modeli su modeli kod kojih je hipoteza ovako definirana:

$$h(\mathbf{x}) = f(\mathbf{w}^\top \phi(\mathbf{x})). \quad (4.15)$$

U odnosu na linearne modele, oni još imaju **prijenosnu (aktivacijsku)** funkciju $f: \mathbb{R} \rightarrow \mathbb{R}$. Ako je f nelinearna, model je nelinearan u parametrima, ali granica klasifikacijskog modela je i dalje hiperravnina.

Slijedi pregled nekih linearnih modela prema Šnajder (2017):

1. Linearna regresija:

$$\begin{aligned} h(\mathbf{x}; \mathbf{w}) &= \mathbf{w}^\top \phi(\mathbf{x}), \\ p(y \mid \mathbf{x}, \mathbf{w}) &= \mathcal{N}(h(\mathbf{x}), \sigma^2)(y), \\ L(y, h(\mathbf{x})) &= (h(\mathbf{x}) - y)^2, \\ \nabla_{\mathbf{w}} L(y, h(\mathbf{x})) &= (h(\mathbf{x}) - y) \phi(\mathbf{x}), \end{aligned}$$

gdje $y \in \mathbb{R}$.

2. Logistička regresija:

$$\begin{aligned} h(\mathbf{x}; \mathbf{w}) &= \sigma(\mathbf{w}^\top \phi(\mathbf{x})) = P(y = 1 \mid \mathbf{x}, \mathbf{w}), \\ P(y \mid \mathbf{x}, \mathbf{w}) &= h(\mathbf{x})^y (1 - h(\mathbf{x}))^{1-y}, \\ L(y, h(\mathbf{x})) &= -y \ln h(\mathbf{x}) - (1 - y) \ln(1 - h(\mathbf{x})), \\ \nabla_{\mathbf{w}} L(y, h(\mathbf{x})) &= (h(\mathbf{x}) - y) \phi(\mathbf{x}), \end{aligned}$$

gdje $y \in \{0, 1\}$, a $\sigma(s) := \frac{1}{1 + \exp(-s)}$ **logistička funkcija**.

3. Višeklasna logistička regresija:

$$\begin{aligned} h(\mathbf{x}; \mathbf{W}) &= \text{softmax}(\mathbf{W} \phi(\mathbf{x})) = [P(y = k \mid \mathbf{x}, \mathbf{w})]_{k=1..C}^\top, \\ P(y \mid \mathbf{x}, \mathbf{w}) &= h_y(\mathbf{x}) = \prod_k h_k(\mathbf{x})^{\mathbb{I}[y=k]}, \\ L(y, h(\mathbf{x})) &= - \sum_k \mathbb{I}[y = k] \ln h_k(\mathbf{x})^{\mathbb{I}[y=k]}, \\ \nabla_{(\mathbf{W}_{[k,:]}^\top)} L(y, h(\mathbf{x})) &= (h_k(\mathbf{x}) - y_k) \phi(\mathbf{x}) \\ \nabla_{\mathbf{W}} L(y, h(\mathbf{x})) &= \phi(\mathbf{x})^\top (h(\mathbf{x}) - \mathbf{e}_y), \end{aligned}$$

gdje $y \in \{1..C\}$, $\text{softmax}(\mathbf{s}) := \frac{1}{\mathbf{1}^\top \exp(\mathbf{s})} \exp(\mathbf{s})$, $h(\mathbf{x}) = [h_k(\mathbf{x})]_{k=1..C}^\top$, $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$, a \mathbf{e}_k označava jednodjinični vektor (vektor kanonske baze) s elementima $\mathbf{e}_{\mathbf{k}[i]} := \mathbb{I}[i = k]$.

Funkcije gubitka su definirane kao negativni logaritam izglednosti, $L(y, h(\mathbf{x})) = -\ln P(y \mid \mathbf{x}, \mathbf{w})$, i konveksne su. Optimalne težine linearne regresije mogu se analitički izračunati, logistička regresija i višeklasna logistička regresija se obično uče optimizacijskim postupcima temeljenim na gradijentu.

Razdiobe $P(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ poopćenih linearnih modela spadaju u **ekspnencijalnu familiju razdioba**. Može se pokazati da je to jedina familija razdioba za koje postoje konjugatne apriorne razdiobe, što pojednostavljuje računanje aposteriorne razdiobe (Murphy, 2012). Opći oblik ekponencijalne familije i više o njenim svojstvima i svojstvima poopćenih linearnih modela može se naći u Murphy (2012).

5. Duboko učenje i konvolucijske mreže

Na ovaj odjeljak imaju utjecaj [Goodfellow et al. \(2016\)](#) i predavanja iz predmeta *Duboko učenje*.

Klasični (plitki) modeli strojnog učenja (npr. poopćeni linearni modeli) oslanjaju se na kvalitetne značajke, tj. funkciju ϕ koja transformira ulazne primjere u vektore značajki. Za neke zadatke koji uključuju visokodimenzionalne primjere sa složenom strukturom (npr. slike, tekst i zvuk) ručno konstruiranje transformacije koja bi bila dovoljno dobra nije izvedivo. Ni jezgrene metode kod kojih se preslikavanje temelji na pretpostavci sličnosti primjera bliskih u ulaznom prostoru ne generaliziraju dobro zbog **prokletstva dimenzionalnosti** ([Bengio et al., 2005](#)). Kod **dubokog učenja** ([LeCun et al., 2015](#); [Goodfellow et al., 2016](#)) transformacija ϕ se uči.

Odabirom

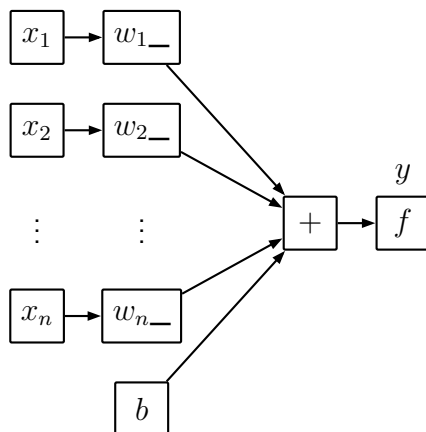
$$\phi(\mathbf{x}) = \phi(\mathbf{x}; \boldsymbol{\theta}_h) = f(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h), \quad (5.1)$$

gdje je \mathbf{W}_h matrica težina, \mathbf{b}_h vektor pomaka, $\boldsymbol{\theta}_h = (\mathbf{W}_h, \mathbf{b}_h)$ a f nelinearna prijenosna (aktivacijska) funkcija koja se primjenjuje na svaki element vektora posebno, dobiva se jednostavna **umjetna neuronska mreža** (ovdje će se koristiti kraći nazivi: *neuronska mreža* ili *mreža*) s jednim **skrivenim slojem** kojem odgovara funkcija ϕ . Ako to uvrstimo u jednadžbu poopćenog linearnog modela (jednadžba (4.15)):

$$h(\mathbf{x}; \boldsymbol{\theta}) = f(\mathbf{w}^T f(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h) + b), \quad (5.2)$$

ili, ako je izlaz vektor,

$$h(\mathbf{x}; \boldsymbol{\theta}) = f(\mathbf{W}_o f(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h) + \mathbf{b}_o), \quad (5.3)$$



Slika 5.1: Grafički prikaz umjetnog neurona. $w_$ označava da se u w množi s ulazom čvora.

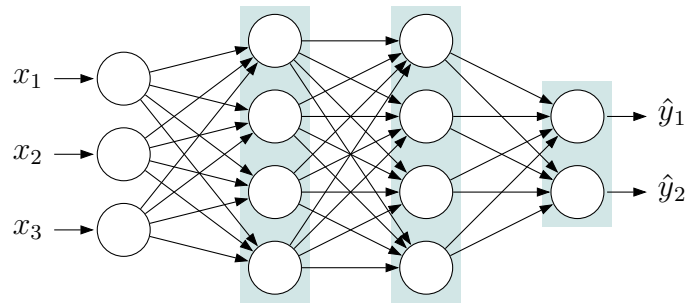
gdje $\theta = (\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_o, \mathbf{b}_o)$. Jedinice neuronske mreže kojima odgovaraju operacije oblika $\mathbf{x} \mapsto f(\mathbf{w}_i^T \mathbf{x} + b_i)$ nazivaju se **umjetni neuroni**. Uz taj naziv, ovdje će se još koristiti naziv **jedinica**. Model umjetnog neurona prikazan je na slici 5.1.

Za razliku od modela opisanih u odjeljku 4.7, za ovakav i dublje modele opisane u sljedećim odjeljcima ciljna funkcija nije konveksna (ni unimodalna), pa nije garantirano da će postupak učenja pronaći dobru hipotezu. Empirijski rezultati ipak pokazuju da duboke mreže uz neke prilagodbe uspješno uče i generaliziraju. Algoritmi koji se koriste za učenje modela dubokog učenja temelje se na gradijentnom spustu. Oni su opisani u odjeljku 5.2.

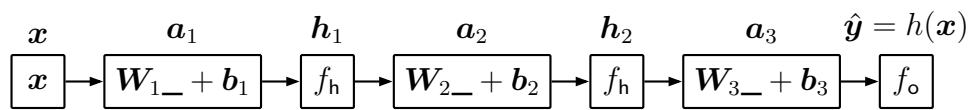
5.1. Duboke unaprijedne mreže

Može se pokazati da model mreže s jednim skrivenim slojem opisan jednačbom (5.3), ako je dimenzija skrivenog sloja dovoljno velika, može s proizvoljno malom greškom aproksimirati svaku neprekinutu funkciju kojoj je domena konveksni podskup od \mathbb{R} . O tome govori teorem o univerzalnoj aproksimaciji (Cybenko, 1989; Leshno et al., 1993). Aktivacijska funkcija mora biti nelinearna jer kompozicija linearnih funkcija je linearna funkcija. Teorem o univerzalnoj aproksimaciji ne govori o tome hoće li takav model generalizirati. Dodavanjem jedinica u skriveni sloj povećava se kapacitet modela.

Obična neuronska mreža može imati više skrivenih slojeva, što se može prikazati kao na slici 5.2 ili apstraktnije, kao na slici 5.3. Povećavanjem broja skrivenih slojeva svaka jedinica u nekom sloju može koristiti izlaze svih jedinica prethodnog



Slika 5.2: Prikaz primjera troslojne mreže. Svakom bridu odgovara jedna težina (pomaci nisu prikazani). Slojevi su označeni plavim četverokutima. Čvorovi koji su unutar četverokuta mreže predstavljaju umjetne neurone. Slika je napravljena prema <http://www.texample.net/tikz/examples/neural-network/>.



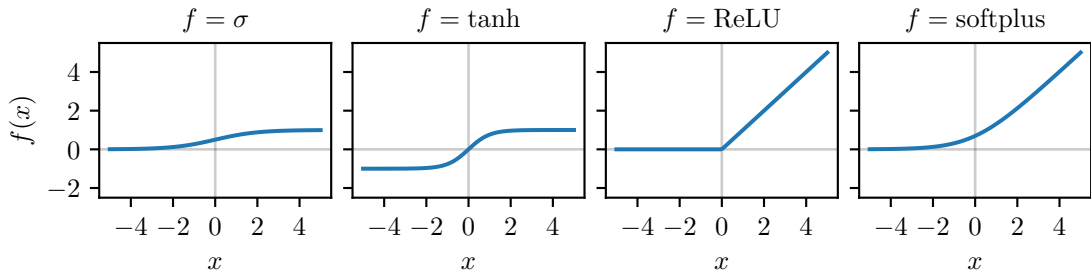
Slika 5.3: Prikaz troslojne mreže kao računskog grafa. Čvorovi predstavljaju funkcije s parametrima, a bridovi podatke (vektore) čije su oznake prikazane uz neke od čvorova iz kojih izlaze. Funkcije su označene oznakom funkcije (aktivacijska funkcija) ili definicijom (afina transformacija). Ulaz sloja označen je s $\underline{}$, a oznake varijabli koje pripadaju čvorovima (ulaz u ulaznom čvoru i parametri u čvorovima afine transformacije) nisu podvučene.

sloja kao značajke, što mreži omogućuje da, u odnosu na mrežu s 1 skrivenim slojem, s manje jedinica modelira funkcije u kojima postoje uzorci koji se ponavljaju i imaju hijerarhijsku strukturu (Goodfellow et al., 2016). Posebne vrste dubokih modela koji uz to iskorištavaju još neke pretpostavke su zato jako uspješne u zadacima u vezi slika, zvuka, teksta i drugih signala. Niži slojevi služe višim slojevima kao značajke transformiranjem kojih se dobivaju značajke više razine.

Kao prijenosna funkcija skrivenih slojeva često se koristi **zglobnica** (**ReLU**, engl. rectified linear unit) $\text{ReLU}(x) = \max(0, x)$ za koju se empirijski pokazalo da ima prednosti nad funkcijama koju su prije bile češće korištene (Glorot et al., 2011). Prije su češće bile korištene **logistička funkcija**, $\sigma(s) = \frac{\exp(s)}{1 + \exp(s)}$, i **tangens hiperbolni**, $\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$. Na slici 5.4 prikazani su grafovi nekih prijenosnih funkcija.

U izlaznom sloju obično se koriste funkcije korištene kod poopcenih linearnih modela (odjeljak 4.7) uz istu probabilističku interpretaciju — identitet za regresiju, logistička funkcija za binarnu klasifikaciju, a softmaxs, $\text{softmax}(s) = \frac{1}{\mathbf{1}^\top \exp(s)} \exp(s)$, koji kao izlaz daje normalizirani vektor koji predstavlja razdiobu, za višeklasnu klasifikaciju.

Dosad opisivane mreže nazivaju se **unaprijedne mreže** (engl. *feedforward*



Slika 5.4: Primjeri prijenosnih funkcija.

networks) zato što se pri izračunu informacija propagira od ulaza prema izlazu, bez povratnih veza. Za mrežu kažemo da je duboka ako ima veći broj slojeva. Struktura duboke unaprijedne mreže ne mora se sastojati samo od niza afinih transformacija i nelinearnosti. Općenito, mrežu možemo predstaviti **računskim grafom**, tj.

usmjerenim acikličkim grafom kod kojega čvorovi predstavljaju varijable ili računske operacije i njihove izlaze, a bridovi označavaju ovisnosti, tj. koji čvor je ulaz kojeg čvora. Svaki čvor koji nije ulazni čvor predstavlja funkciju koju ostvaruje podgraf koji čine njegovi preci, pa ga možemo poistovjetiti s funkcijom čiji su parametri svi ulazni čvorovi iz skupa čvorova predaka. Čvorovi koji nemaju roditelje su varijable koje čine ulazi i parametri. Parametri se mogu dijeliti, tj. mogu biti ulaz većem broju čvorova kao i svi drugi čvorovi. Čvorovi koji nemaju djecu su izlazi računskog grafa. Na slici 5.1 i 5.3 su prikazani takvi grafovi s različitim razinama apstrakcije. U njima, radi sažetosti, parametri nemaju zasebne čvorove, nego su označeni unutar čvorova koji o njima ovise.

5.2. Učenje

Cilj učenja je pronaći parametre θ koji minimiziraju pogrešku

$$E(\theta; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} L(y_i, h(x_i; \theta)) + \lambda R_R(\theta) \quad (5.4)$$

i postići dobru generalizaciju. Duboki modeli se obično uče algoritmima koji se temelje na gradijentnom spustu. Gradijent pogreške s obzirom na parametre je

$$\nabla_{\theta} E(\theta; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \nabla_{\theta} L(y_i, h(x_i; \theta)) + \lambda \nabla_{\theta} R_R(\theta). \quad (5.5)$$

Kod dubokih mreža, tj. usmjerenih acikličkih računskih grafova, gradijent se računa **algoritmom propagacije pogreške unatrag** (Rumelhart et al., 1986) koji se

temelji na **pravilu deriviranja kompozicije funkcija i dinamičkom programiranju**.

U ovom odjeljku su kratko opisane ideje korištene za efikasno računanje gradijenta i optimizacijski postupci koji se koriste za pronalaženje dobrih parametara kod dubokih mreža.

5.2.1. Algoritam propagacije pogreške unatrag

Na ovaj pododjeljak ima utjecaj [Olah \(2015a\)](#).

Gradijent gubitka nije potrebno analitički računati za svaki parametar posebno. Primjenom pravila deriviranja složene funkcije, derivacija vrijednosti nekog čvora b s obzirom na čvor vrijednost nekog čvora $a \in \text{pred}_G(b)$ jednaka je zbroju umnožaka parcijalnih derivacija svakog djeteta s obzirom na roditelja po svakom putu između a i b . Pri tome je put definiran kao niz takav da sljedeći (usmjereni) brid počinje u čvoru u kojem je prethodni završio. Svakom bridu (p, c) odgovara derivacija $\frac{\partial c}{\partial p}$. Derivacije između susjednih čvorova ne moraju se računati za svaki put posebno. Već izračunate derivacije se mogu ponovo koristiti. Isto vrijedi ako su vrijednosti čvorova vektori (ako su višedimenzionalni nizovi, možemo ih svesti na vektore) i ako računamo Jakobijeve matrice. Dalje će se za Jakobijeve matrice isto koristiti riječ *derivacija*.

Algoritam propagacije pogreške unatrag se tako naziva zato što se izračun gradijenta širi od izlaznog čvora (ili čvora koji predstavlja gubitak ili funkciju pogreške) prema njegovim roditeljima, pa prema roditeljima roditelja itd. uz primjenu pravila deriviranja kompozicije funkcija. Pri tome se ne moraju računati gradijenti s obzirom na čvorove koji ne ovise o varijablama za koje se računa gradijent.

Neka je L vrijednost čvora gubitka, a θ neki parametar. Želimo izračunati gradijent $\nabla_{\theta_i} L = \frac{\partial L}{\partial \theta}^T$. Derivacija gubitka s obzirom na čvoru u se može rekurzivno izraziti:

$$\frac{\partial L}{\partial u} = \sum_{c \in \text{ch}_G(u)} \frac{\partial L}{\partial c} \frac{\partial c}{\partial u}, \quad (5.6)$$

gdje su c djeca čvora u . Ako $\frac{\partial L}{\partial c}$ nije izračunat za trenutni ulaz, izračuna se, a inače se koristi prethodno izračunata vrijednost. Ista jednadžba vrijedi za čvorove parametara.

Operacija	Derivacije
$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$,	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{W}$, $\frac{\partial \mathbf{y}_{[i]}}{\partial (\mathbf{w}_{[j,:]}^\top)} = \llbracket i = j \rrbracket \mathbf{x}^\top$, $\frac{\partial \mathbf{y}}{\partial \mathbf{b}} = \mathbf{I}$
$\mathbf{y} = \mathbf{a} \odot \mathbf{b}$,	$\frac{\partial \mathbf{y}}{\partial \mathbf{a}} = \text{diag}(\mathbf{b})$, $\frac{\partial \mathbf{y}}{\partial \mathbf{b}} = \text{diag}(\mathbf{a})$
$\mathbf{y} = \text{ReLU}(\mathbf{x})$	$\frac{\partial \mathbf{y}_{[i]}}{\partial \mathbf{x}_{[j]}} = \llbracket i = j \rrbracket \llbracket \mathbf{x}_{[j]} > 0 \rrbracket$
$\mathbf{y} = \sigma(\mathbf{x})$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \text{diag}(\mathbf{y} \odot (\mathbf{1} - \mathbf{y}))$
$\mathbf{y} = \tanh(\mathbf{x})$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \text{diag}(1 - \mathbf{y} \odot \mathbf{y})$
$\mathbf{y} = \text{softmax}(\mathbf{x})$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_{[j]}} = \mathbf{y} \odot (\mathbf{e}_j - \mathbf{y})$
$y = -t \ln \sigma(x) - (1 - t) \ln(1 - \sigma(x))$	$\frac{\partial y}{\partial x} = \sigma(x) - t$
$y = -\ln \text{softmax}(\mathbf{x})_{[j]}$	$\frac{\partial y}{\partial \mathbf{x}} = (\text{softmax}(\mathbf{x}) - \mathbf{e}_j)^\top$

Tablica 5.1: Parcijalne derivacije (Jakobijeve matrice) nekih operacija po njihovim ulazima. Zadnja sva retke predstavljaju gubitak unakrsne entropije (negativni logaritmi izglednosti) za binarnu i višeklasnu klasifikaciju, gdje je t indeks ciljne klase. \mathbf{e}_t označava jednojedinичni vektor s elementima $\mathbf{e}_{j[i]} := \llbracket i = j \rrbracket$.

Neka je zadatak nadzirano učenje, $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_i$ skup podataka za učenje, a $L_i = L(\mathbf{y}_i, h(\mathbf{x}_i, \boldsymbol{\theta}))$ gubitak para $(\mathbf{x}_i, \mathbf{y}_i)$. Neka je pogreška npr. $E = \sum_i L_i + R_{\mathbf{R}}(\boldsymbol{\theta})$. Onda

$$\frac{\partial E}{\partial \boldsymbol{\theta}} = \sum_i \frac{\partial L_i}{\partial \boldsymbol{\theta}} + \frac{\partial R_{\mathbf{R}}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}), \quad (5.7)$$

gdje se izrazi na desnoj strani računaju rekurzivno uz pamćenje izračunatih gradijenata (ili unaprijed izračunate gradijente koji odgovaraju bridovima u podgrafu koji se sastoji od čvorova potomaka) prema jednadžbi 5.6.

5.2.2. Gradijenti nekih osnovnih operacija

U tablici 5.1 prikazane su parcijalne derivacije (Jakobijeve matrice) nekih operacija s obzirom na njihove ulaze. Korištenjem pravila deriviranja kompozicije funkcija mogu se izračunati derivacije složenijih funkcija. Radi efikasnosti se izračunavanje vrijednosti u računskom grafu i algoritam propagacije pogreške unatrag provodi paralelno za više ulaza odjednom. Izvodi gradijenata nekih operacija uz višestruke ulaze mogu vidjeti npr. ovdje: <http://www.zemris.fer.hr/~ssegvic/du/lab0.shtml>.

5.2.3. Stohastička optimizacija

U pododjeljku 2.3.1 opisan je gradijentni spust i neki izvedeni algoritmi koji koriste neke dodatne heuristike. U ovom pododjeljku opisana je primjena tih algoritama u dubokom učenju. Kod učenja dubokih modela se obično koristi puno podataka i iteracija optimizacije se provodi procjenjivanjem gradijenta funkcije pogreške na temelju manjeg dijela skupa za učenje.

Kod **stohastičkog gradijentnog spusta** se u nekoj iteraciji gradijentnog spusta umjesto gradijenta pogreške koristi gradijent procjene pogreške na temelju nekog podskupa skupa za učenje ili samo jednog primjera. Takav algoritam naziva se **stohastički gradijentni spust**. Moguće je podskupove u svakoj iteraciji ponovo slučajno izvlačiti iz cijelog skupa za učenje \mathcal{D} , ali obično se iteracije podijele na **epohe** od kojih se svaka sastoji od B iteracija, u svakoj epohi se skup za učenje slučajno podijeli na B nepreklapajućih podskupova \mathcal{D}_i jednake veličine, od kojih se svaki koristi u jednoj iteraciji unutar epohe. Skupove \mathcal{D}_i nazivamo **mini-grupe**. U iteraciji i u nekoj epohi koristi se procjena gradijenta

$$\nabla_{\theta} E(\theta; \mathcal{D}_i) = \frac{1}{|\mathcal{D}_i|} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_i} \nabla_{\theta} L(\mathbf{y}_i, h(\mathbf{x}_i; \theta)) + \lambda \nabla_{\theta} R_R(\theta). \quad (5.8)$$

i iteracija (prema jednadžbi (2.64)) ima oblik

$$\theta_i = \theta_{i-1} - \eta \nabla_{\theta} E(\theta; \mathcal{D}_i), \quad (5.9)$$

gdje je e broj epohe, a $i + 1$ broj trenutne iteracije unutar epohe.

Prema Goodfellow et al. (2016), na odabir veličine mini-grupe utječu:

1. Kvaliteta procjene gradijenta. Veće minigrupe daju točniju procjenu gradijenta.
2. Računska efikasnost. Premale mini-grupe ne iskorištavaju potpuno mogućnost paralelizacije izračuna, a prevelike grupe ne stanu u memoriju.
3. Optimizacija s manjim mini-grupama ima učinak regularizacije (Wilson i Martinez, 2003), ali zahtijeva manju stopu učenja i sporije konvergira.

Kako bi optimizacijski algoritam konvergirao, treba se smanjivati stopa učenja ovisno o iteraciji (epohi). Prema Goodfellow et al. (2016), dovoljan uvjet za

konvergenciju gradijentnog spusta je

$$\sum_{k=1}^{\infty} \eta_k = \infty \wedge \sum_{k=1}^{\infty} \eta_k^2 < \infty, \quad (5.10)$$

gdje je k broj iteracije od početka učenja.

Kako bi se ublažio šum procjene gradijenta i ubrzalo učenje, obično se upotrebljava inercija, kao što je opisano u pododjeljku 2.3.1. Empirijski se pokazuje da stohastički gradijentni spust s momentom postiže dobru generalizaciju. U pododjeljku 2.3.1 su opisana i dva algoritma koja koriste pokretne prosjeke momenata gradijenta i prilagođeni su stohastičkom učenju s mini-grupama. RMSProp skalira gradijent po elementima korištenjem pokretnog prosjeka kvadrata gradijenta tako da norma elemenata pomaka ne ovisi jako o prosječnoj normi gradijenta u zadnjim iteracijama. Adam koristi inerciju i obavlja skaliranje slično kao RMSProp.

5.2.4. Inicijalizacija parametara

Kod učenja dubokih modela jako je bitna inicijalizacija parametara. Sve težine mreže (ili dijela nje), npr. kao na slici 5.2, se ne smiju se inicijalizirati konstantnom vrijednošću. Zamjenom dvaju jedinica istog sloja, npr. kao na slici 5.2, dobiva se ista mreža i gradijent je jednak za sve težine unutar istog sloja, osim za zadnji sloj. To se rješava inicijalizacijom težina nasumičnim vrijednostima. Ako su inicijalizirane težine manje, sporije će se *razbijati* simetrija, a ni prevelike težine nisu dobre. Ako se koriste prijenosne funkcije sa zasićenjem problem mogu biti težine s prevelikom apsolutnim vrijednostima jer mogu uzrokovati zasićenje i tako onemogućavati učenje. Taj problem rješava zglobnica, ali množenjem velikih težina kroz više slojeva daje sve veće izlaze, što kod linearnih slojeva daje prevelik gradijent, što se vidi u tablici 5.1.

Heuristike korištene za inicijalizaciju težina se temelje na aproksimiranju mreže nizom matričnih množenja i postizanju da varijance gradijenata (i izlaza) budu otprilike konstantne kroz mrežu (Goodfellow et al., 2016). Otprilike konstantna varijanca gradijenta može se ostvariti inicijalizacijom slučajnim vrijednostima iz Guassove ili unifomne razdiobe s varijancom $\frac{1}{n}$, gdje je n broj ulaza. Glorot i Bengio (2010) kao kompromis između jednake varijance gradijenta i jednake varijance izlaza slojeva predlažu varijancu $\frac{1}{n+m}$, gdje je m broj izlaza (Goodfellow et al., 2016).

Pomaci se obično inicijaliziraju na neku konstantu.

5.2.5. Problem nekonveksnosti funkcije pogreške

Goodfellow et al. (2016) navode sljedeće probleme koji se javljaju kod nekonveksne optimizacije:

1. **Loše kondicioniranje Hesseove matrice.** Loše kondicioniranje Hesseove matrice može biti razlog da i s jako malim korakom učenje funkcija pogreške raste u smjeru gradijenta zato što kvadratni član u Taylorovom razvoju u jednadžbi (2.65) bude pozitivan i prevlada linearni član.
2. **Lokalni minimumi.** I ako se zanemare ekvivalentni lokalni minimumi koji postoje zbog simetričnosti zamjenjivosti položaja neurona u istom sloju i drugih simetričnosti u neuronskim mrežama, funkcija pogreške ima velik broj lokalnih minimuma. Empirijski se pokazuje da loši lokalni minimumi nisu problem i da nije potrebno pronaći globalni minimum kako bi se dobili dobri rezultati.
3. **Ostale stacionarne točke.** Kod visokodimenzionalnih optimizacijskih problema lokalni minimumi i maksimumi su obično rijetki zato što bi onda sve vlastite vrijednosti Hesseove matrice morale biti istog predznaka. Zato su češće sedlaste točke kod kojih se predznak barem jedne vlastite vrijednosti razlikuje od predznaka ostalih. Empirijski se pokazuje da sedlaste točke kod dubokih mreža nisu velik problem za optimizacijske postupke prvog reda koje ne privlače sedlaste točke. I ako se parametri nalaze točno u sedlastoj točki tako da je gradijent 0, stohastički gradijentni spust može imati drugačije gradijente.
4. **Litice i eksplodirajući gradijenti.** Kod nekih modela javlja se problem velikih vrijednosti gradijenta u nekim točkama. To se može riješiti ograničavanjem norme gradijenta.

5.3. Regularizacija i poboljšavanje učenja

U ovom pododjeljku opisani su neki od češćih postupaka koji se koriste za poboljšavanje učenja, tj. postizanja bolje generalizacije i bržeg učenja. Dijelovi ovog

pododjeljka temelje se na [Goodfellow et al. \(2016\)](#), gdje se može naći opširniji i dublji pregled.

5.3.1. Kažnjavanje norme težina

Najjednostavniji način regularizacije je kažnjavanje norme težina. Regularizacijski dio funkcije pogreške R_R se najčešće definira kao kvadrat L^2 norme, tj. koristi se L^2 **regularizacija** koja odgovara apriornoj pretpostavci Gaussove razdiobe težina s dijagonalnom kovarijacijskom matricom i očekivanjem \mathbf{o} . Može se koristiti i L^1 **regularizacija** koja potiče rijetkost težina, tj. postavlja minimum funkcije pogreške u ovisnosti o nekim težinama točno u 0. To je detaljnije objašnjeno npr. u [Goodfellow et al. \(2016\)](#). L^1 regularizacija odgovara Laplaceovoj apriornoj razdiobi. Općenito, gubitak L^p regularizacije ima oblik:

$$R_R(\boldsymbol{\theta}) = \frac{\lambda}{p} \|\boldsymbol{\theta}\|_p^p = \frac{\lambda}{p} \sum_i |\boldsymbol{\theta}_{[i]}|^p, \quad (5.11)$$

gdje λ određuje jačinu regularizacije, tj. koncentraciju apriorne razdiobe. Gustoći apriorne razdiobe odgovara

$$p(\boldsymbol{\theta}) = \frac{1}{Z} \exp(-R_R(\boldsymbol{\theta})) \quad (5.12)$$

$$= \frac{1}{Z} \prod_i \exp\left(-\frac{\lambda}{p} |\boldsymbol{\theta}_{[i]}|^p\right), \quad (5.13)$$

gdje je Z normalizacijska konstanta. Gradijent regularizacijskog gubitka s obzirom na $\boldsymbol{\theta}_{[i]}$ je $\lambda \operatorname{sgn}(\boldsymbol{\theta}_{[i]}) |\boldsymbol{\theta}_{[i]}|^{p-1}$. Posebno, to je $\lambda \boldsymbol{\theta}_{[i]}$ za $p = 2$ i $\lambda \operatorname{sgn}(\boldsymbol{\theta}_{[i]})$ za $p = 1$.

5.3.2. Rano zaustavljanje učenja

Regularizacijski učinak koji se može usporediti s L^p regularizacijom ima **rano zaustavljanje** učenja zato što ograničava koliko se parametri mogu udaljiti od početne vrijednosti. Ako se model predugo uči, može se dogoditi da se težine modela s velikim kapacitetom previše prilagode skupu za učenje i zato dođe do loše generalizacije.

5.3.3. Generiranje podataka

Postupci koji značajno mogu utjecati na generalizaciju su postupci **proširivanja skupa podataka**, što znači da se tijekom učenja obično provode jednostavne slučajne transformacije nad primjerima prije nego što se daju kao ulaz modelu. Primjeri transformacija koje se mogu koristiti u računalnom vidu, ovisno o zadatku, su reflektiranje, translacija i rotacija. Generiranje podataka kod zadataka koji imaju veze sa zvukom isto može biti korisno [Goodfellow et al. \(2016\)](#).

Dodavanje šuma ulazu isto može biti korisno ([Goodfellow et al., 2016](#)). To odgovara pretpostavci da primjeri koji su slični u ulaznom prostoru trebaju biti slični i u izlaznim prostoru.

5.3.4. Isključivanje neurona — dropout

Dropout ([Hinton et al., 2012](#); [Srivastava et al., 2014](#)) je postupak regularizacije koji unosi šum u izlaze skrivenih slojeva tijekom učenja. Obično se ostvaruje tako da se tijekom učenja za svaki primjer svaka jedinica mreže isključi s vjerojatnošću $1 - p$, koja je hiperparametar. Tijekom testiranja, tj. zaključivanja, sve se jedinice skaliraju s p , tj. očekivanjem skaliranja koje je tijekom učenja 0 s vjerojatnošću $1 - p$, a 1 s vjerojatnošću p . *Dropout* se obično primjenjuje nakon afine transformacije (ne nakon aktivacije).

Učenje s *dropoutom* se može interpretirati kao učenje eksponencijalnog broja modela koji dijele parametre, a zaključivanje kao aproksimacija usrednjavanja modela ili aproksimacija bayesovskog zaključivanja. Vremenski zahtjevniji, ali ispravniji postupak usrednjavanja modela bio bi uzorkovanje ([Srivastava et al., 2014](#)), tj. *Monte Carlo* aproksimacija izlaza. [Gal i Ghahramani \(2015a\)](#) su dali bayesovsku interpretaciju takvog usrednjavanja.

Umjesto Bernoullijeve razdiobe, skaliranje ili izlazi jedinica mogu imati npr. Gaussovu razdiobu ili neku drugu.

5.3.5. Normalizacija po grupama

Normalizacija po grupama (engl. *batch normalization*) ([Ioffe i Szegedy, 2015](#)) je postupak koji ublažava probleme pri učenju i omogućuje uspješno učenje jako dubokih modela. Prema [Goodfellow et al. \(2016\)](#), problem kod učenja jako dubokih

modela je što se se sastoje od kompozicije velikog broja funkcija, zbog čega je velika međuzavisnost između parametara različitih slojeva, a u koraku gradijentnog spusta parametri svih funkcija ažuriraju se istovremeno. Gradijenti spust pretpostavlja da je utjecaj svakog parametra lokalno nezavisan, tj. svaka se funkcija (sloj afine transformacije) prilagođava ostatku mreže kakav je u trenutnom koraku, tj. očekuje da se prethodni slojevi neće promijeniti.

U Goodfellow et al. (2016) je to objašnjeno na pojednostavljenom primjeru $\hat{y} = xw_1w_2, \dots, w_j$, gdje su elementi gradijenta $g_i = \nabla_{w_i}\hat{y} = x \prod_{j \neq i} w_j$. Novi izlaz nakon koraka gradijentnog spusta je $x \prod_i (w_i - \epsilon g_i)$ gdje članovi uz više potencije ϵ mogu imati utjecaj koji raste eksponencijalno s dubinom l .

Sloj normalizacije po grupama se dodaje nakon sloja linearne transformacije (prije prijenosne funkcije). On kod učenja obavlja ovakvu operaciju:

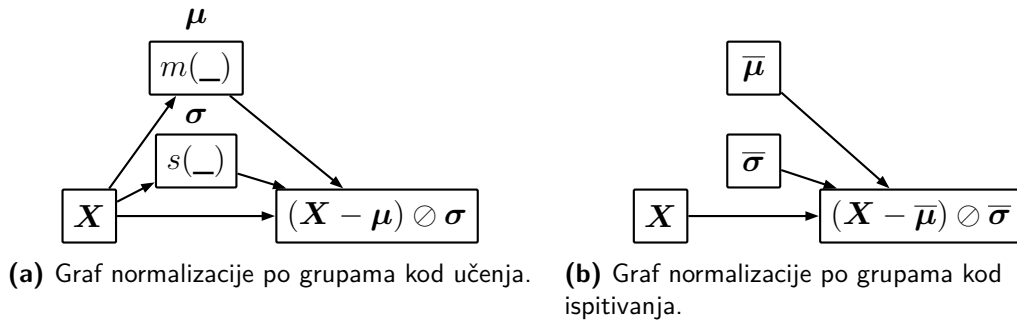
$$\mathbf{Y} = (\mathbf{X} - m(\mathbf{X})) \oslash s(\mathbf{X}), \quad (5.14)$$

gdje je $\mathbf{X} = [\mathbf{x}_i]_{i=1..N}^T \in \mathbb{R}^{N \times n}$ matrica kojoj su reci vektori značajki pojedinih primjera, $\mathbf{Y} = [\mathbf{y}_i]_{i=1..N}^T \in \mathbb{R}^{N \times n}$ matrica kojoj su reci značajke normalizirane ulazne značajki, $m(\mathbf{X}) := \frac{1}{N} \sum_i \mathbf{X}_{[i,:]} \in \mathbb{R}^{1 \times n}$ srednja vrijednost vektora značajki, $s(\mathbf{X}) := \left(\frac{1}{N} \sum_i (\mathbf{X}_{[i,:]} - m(\mathbf{X}))^{\odot 2} \right)^{\odot \frac{1}{2}} \in \mathbb{R}^{1 \times n}$ standardna devijacija vektora značajki po elementima. Oduzimanje u jednadžbi (5.14) je definirano tako da se od svakog retka \mathbf{X} oduzima $m(\mathbf{X})$. Takvo značenje ima i dijeljenje. Izlaz sloja normalizacije po grupama tijekom učenja je invarijantan na skaliranje i pomak ulaza \mathbf{X} , kao i skaliranje težina linearne transformacije koja prethodi normalizaciji po grupama.

Statistike grupe, tj. srednje vrijednosti i standardne devijacije od grupe za koju se provodi zaključivanje, se koriste samo kod učenja. Inače se koriste statistike skupa za učenje koje se mogu procjenjivati pokretnim prosjekom tijekom učenja. Računski graf normalizacije po grupama kod učenja i kod ispitivanja je prikazan na slici 5.5.

Kako se ne bi izgubila ekspresivnost, nakon sloja normalizacije po grupama obično se dodaje pomak $\beta \in \mathbb{R}^{1 \times n}$ i skaliranje $\gamma \in \mathbb{R}^{1 \times n}$ svake značajke. β i γ su parametri koji se uče. Kod ispitivanja je normalizacija po grupama uz skaliranje i pomak samo drugačija parametrizacija koja je uz prethodni sloj linearne transformacije s težinama \mathbf{W} može svesti na sloj afine transformacije s težinama $\mathbf{W} \oslash \sigma^T \odot \gamma^T$ i pomacima $-\mu \oslash \sigma \odot \gamma + \beta$.

Kod konvolucijskih mreža se normalizacija po grupama ne provodi samo po



Slika 5.5: Grafovi normalizacije po grupama kod učenja i kod ispitivanja. m i s su funkcije koje računaju srednju vrijednost i standardnu devijaciju grupe. $\bar{\mu}$ je srednja vrijednost, a $\bar{\sigma}$ standardna devijacija ulaza kod skupa za učenje.

dimenziji grupe, nego i po dimenzijama konvolucije, po kojima treba vrijediti translacijska ekvivarijantnost. Npr. ako je ulazni niz dimenzija $N \times H \times W \times C$, gdje je N veličina grupe, H visina slike, W širina slike, a C broj značajki, tj. broj filtara zadnje konvolucije, vektor srednjih vrijednosti i vektor standardnih devijacija je dimenzije C , tj. $1 \times 1 \times 1 \times C$.

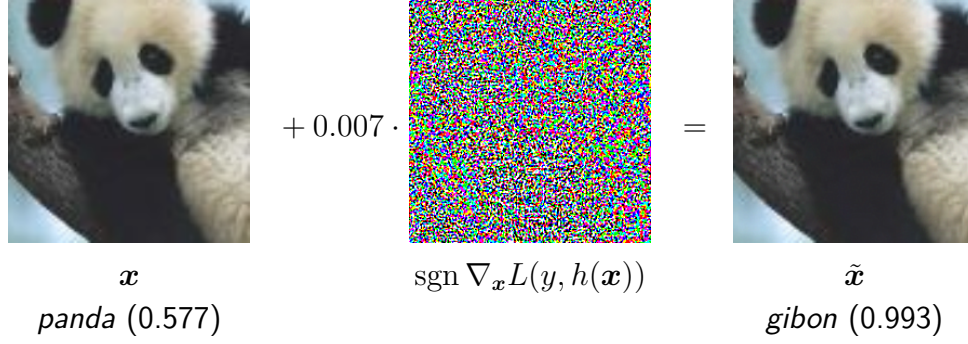
Normalizacija po grupama i regularizacija

Pokazuje se da normalizacija po grupama ima i regularizacijski učinak, vjerojatno zbog stohastičnosti mini-grupa i same reparametrizacije. Zbog neovisnosti izlaza normalizacijskog sloja o skaliranju težina linearnog sloja koji mu prethodi tijekom učenja, L^2 regularizacija nema regularizacijski utjecaj na linearne slojeve, ali ima na veličinu koraka učenja u odnosu na težine (van Laarhoven, 2017). Pokazuje se da *dropout* uz normalizaciju po grupama često ima slab ili negativan učinak na učenje. Li et al. (2018) to objašnjavaju manjom varijancom tijekom testiranja u odnosu na varijancu tijekom učenja i predlažu izmjene za smanjivanje negativnog učinka.

5.3.6. Neprijateljski primjeri i regularizacija za postizanje otpornosti na njih

Ovaj odjeljak se temelji na Grubišić (2018).

Pokazalo se da se mogu pronaći ulazni primjeri (npr. slike) koji su u ljudskoj percepciji slični prirodnim primjerima, ali i modeli koji na neizmjenjenim primjerima ostvaruju rezultate usporedive s rezultatima ljudi daju krive predikcije (Szegedy et al., 2013; Goodfellow et al., 2014). Takvi ulazni primjeri se nazivaju



Slika 5.6: Prilagođeni prikaz dobivanja neprijateljskog primjera FGSM-om iz [Goodfellow et al. \(2014\)](#). Nakošene riječi predstavljaju klase, a brojevi u zagradama vjerojatnosti koje im mreža dodjeljuje.

neprijateljski primjeri. Oni se mogu dobiti i pomoću jednog koraka gradijentnog spusta pomicanjem ulaznog primjera u smjeru povećavanja gubitka. Na slici 5.6 je prikazano generiranje neprijateljskog primjera malom izmjenom izvorne slike.

Neka je $d : \mathbb{X} \times \mathbb{X}$ funkcija udaljenosti u ulaznom prostoru. Za svaki primjer x možemo definirati susjedstvo $B_\epsilon(x) = \{x' : d(x', x) \leq \epsilon\}$. Pronalaženje neprijateljskih primjera se može definirati kao optimizacijski problem pronalaženja primjera \tilde{x} koji maksimizira gubitak uz ograničenje da se nalazi u susjedstvu B_ϵ prirodnog primjera x :

$$\tilde{x} = \arg \max_{\tilde{x} \in B_\epsilon(x)} L(y, h(\tilde{x})). \quad (5.15)$$

Za funkciju udaljenosti d se obično uzima neka L^p -norma razlike. Npr. za L^∞ -normu je $B_\epsilon(x) = \{\tilde{x} : \|\tilde{x} - x\|_\infty \leq \epsilon\}$.

Neprijateljski primjeri mogu se pronaći iterativnim optimizacijskim postupcima prvog reda uz održavanje ograničenja susjedstva. Pokazuje se da je neprijateljske primjere moguće pronaći već samo jednim korakom u smjeru predznaka gradijenta. Jedna vrsta takvog napada je fast gradient sign method (FGSM) [Goodfellow et al. \(2014\)](#). Neprijateljskom primjeru koji se pronalazi FGSM-om odgovara sljedeći izraz:

$$\tilde{x} = x + \epsilon \text{sgn } \nabla_x L(y, h(x)). \quad (5.16)$$

[Madry et al. \(2017\)](#) definiraju iterativni postupak koji se temelji na FGSM-u i nazivaju ga *projected gradient descent* (PGD):

$$\tilde{x}_i = \Pi_{B_\epsilon(x)}(\tilde{x}_{i-1} + \alpha \text{sgn } \nabla_{\tilde{x}_{i-1}} L(y, h(\tilde{x}_{i-1}))). \quad (5.17)$$

Početni \tilde{x} se bira nasumično unutar $B_\epsilon(x)$. α je veličina koraka optimizacije, a

$\Pi_{B_\epsilon(x)}$ označava projekciju na zatvorenu ϵ -kuglu oko prirodnog primjera x uz L^∞ -normu. Npr. projekcijom vektora v na susjedstvo vektora x , $\Pi_{B_\epsilon(x)}(v) = \arg \min_{v' \in B_\epsilon(x)} \|v' - v\|_\infty$, svakoj komponenti $v_{[i]}$ se dodjeljuje najbliža vrijednost unutar intervala $[x_{[i]} - \epsilon, x_{[i]} + \epsilon]$.

Mogući su i napadi bez uvida u strukturu modela, npr. genetskim algoritmom. Također, pokazuje se da su neprijateljski primjeri u velikoj mjeri prenosivi između različitih modela (Szegedy et al., 2013; Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2016; Liu et al., 2016).

Možemo definirati oblik rizika koji se može nazvati **neprijateljski rizik** (Madry et al., 2017):

$$R_A(\theta; d, \epsilon) = \mathbf{E}_{(x,y) \sim \mathcal{D}} \left(\max_{\tilde{x} \in B_\epsilon(x)} L(y, h(\tilde{x}; \theta)) \right). \quad (5.18)$$

Mali neprijateljski rizik predstavlja dobru lokalnu generalizaciju u susjedstvu prirodnih primjera. Jedan od najuspješnijih postupaka za postizanje otpornosti na neprijateljske primjere je **učenje s neprijateljskim primjerima** (engl. *adversarial training*). Kod učenja s neprijateljskim primjerima skup za učenje se proširuje neprijateljskim primjerima koji se tijekom učenja prilagođavaju parametrima mreže Goodfellow et al. (2014). Umjesto prirodne razdiobe \mathcal{D} u jednadžbi 5.18, koriste se podaci za učenje, tj. empirijska razdioba.

Kurakin et al. (2016) primjećuju da korištenja stvarne ciljne oznaka u gubitku koji se maksimizira kako bi se dobio neprijateljski primjer unosi informacije o pravim oznakama u neprijateljske primjere koji se koriste tijekom učenja, na što se model može prenaučiti i može biti neotporan na neprijateljske primjere dobivene postupcima koji ne koriste stvarnu oznaku. Zato predlažu da se umjesto ciljne oznake u gubitku koji se maksimizira koristi oznaka koja odgovara predikciji modela. Onda izrazu (5.15) odgovara ovaj izraz:

$$\tilde{x} = \arg \max_{\tilde{x} \in B_\epsilon(x)} L \left(\arg \max_k h(x)_{[k]}, h(\tilde{x}) \right), \quad (5.19)$$

što je obično maksimizacija negativnog logaritma najvećeg izlaza softmaks:

$$\tilde{x} = \arg \max_{\tilde{x} \in B_\epsilon(x)} \left(-\ln h(\tilde{x})_{[\arg \max_k h(x)_{[k]}]} \right). \quad (5.20)$$

To je slično virtualnim neprijateljskim primjerima koje su predložili Miyato et al. (2015). Miyato et al. (2015, 2017) definiraju regularizacijski gubitak kojemu odgovara ovakav rizik koji možemo nazvati **rizik lokalne zaglađenosti razdiobe**

(engl. *local distributional smoothness*, *LDS*):

$$R_{\text{LDS}}(\boldsymbol{\theta}; d, \epsilon) = \mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \left(\max_{\tilde{\mathbf{x}} \in B_{\epsilon}(\mathbf{x})} D_{\text{KL}}(\mathbf{y} \mid \mathbf{x}, \hat{\boldsymbol{\theta}} \parallel \mathbf{y} \mid \tilde{\mathbf{x}}, \boldsymbol{\theta}) \right), \quad (5.21)$$

Razlika u odnosu na neprijateljski rizik je u tome što se umjesto KL-divergencije između ciljne oznake i razdiobe predikcije za neprijateljski primjer, ovdje koristi KL-divergencija između razdiobe predikcije za neizmijenjeni primjer i razdiobe predikcije za neprijateljski primjer. $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$, ali $\hat{\boldsymbol{\theta}}$ označava da se parametri ne optimiziraju s obzirom na argument koji zamjenjuje oznaku. Takva regularizacija se može koristiti i u polunadziranom učenju, kada nisu poznate oznake svih primjera u skupu za učenje.

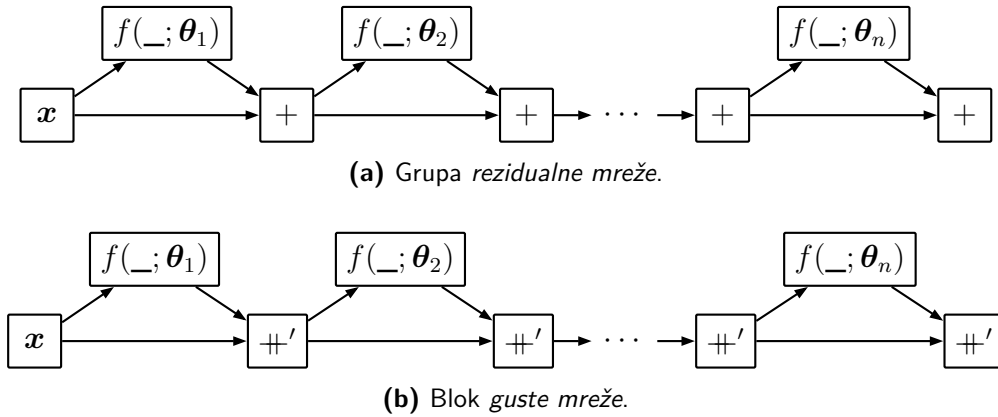
Pokazuje se da se najučinkovitiji neprijateljski primjeri dobivaju u više koraka optimizacije (Kurakin et al., 2016; Madry et al., 2017). Madry et al. (2017) postižu dobru otpornost uz korištenja PGD-a (jednadžba 5.17) za generiranje neprijateljskih primjera tijekom učenja, a zaključuju i da je potreban veći kapacitet kako bi mreža zadržala performansu na prirodnim podacima.

5.3.7. Dijeljenje parametara i dijelova mreže

Dijeljenje ili višestruka uporaba parametara i dijelova mreže je korisna kada se za podatke vrijede neka svojstva ekvivarijantnosti (npr. na pomak u prostoru ili vremenu, skaliranje, kompozitnost,...). Ono se može primjenjivati i kod višezadačnog učenja kod kojeg se dio mreže koristi za dobivanje značajki koje su korisne za učenje većeg broja različitih zadataka.

5.3.8. Pomoćni gubici i preskočne veze

Ponekad kod učenja dubokih mreža mogu pomoći **dodatni gubici** koji ovise o funkcijama koje procjenjuju izlaz na temelju značajki niže razine. Najuspješniji modeli koriste **preskočne veze** koje omogućuju da funkcije koje računaju značajke više razine imaju izravan pristup značajkama niže razine. Najjednostavniji i najčešće korišteni takvi modeli (u računalnom vidu) su *rezidualne mreže* (*ResNet*) (He et al., 2015, 2016) i *guste mreže* (*DenseNet*) (Huang et al., 2016) od kojih je osnovna struktura prikazana na slici 5.7. Empirijski se pokazuje da se povećavanjem dubine takvih mreža poboljšava generalizacija.



Slika 5.7: Osnovne strukture *rezidualnih* i *gustih* mreža. f je obično niz od nekoliko konvolucijskih (ili linearnih slojeva) tako da su ispred svake konvolucije sloj normalizacije po grupama i sloj ReLU-a. $+$ označava zbrajanje, a \oplus' konkatenciju po zadnjoj dimenziji ulaznih nizova.

5.4. Konvolucijske mreže

Konvolucijske mreže su mreže koje, prema definiciji u [Goodfellow et al. \(2016\)](#), na barem jednom mjestu, umjesto općenite linearne transformacije, koriste **konvoluciju**. Konvolucijske mreže koriste pretpostavku **translacijske ekvivarijantnosti** po nekim dimenzijama ulaza i posebno se uspješno primjenjuju na zadacima u vezi slika. Pojedini elementi izlaza **konvolucijskog sloja** računaju se množenjem manjeg **filtra** s elementima ulaza koje on pokriva na svakom položaju ulaza. Element i izlaza ovise o malom broju elemenata ulaza oko odgovarajućih položaja, tj. **povezanost** je **lokalna**. To omogućuje da se broj parametara konvolucijskog sloja značajno smanji u odnosu na **potpuno-povezani sloj**, tj. sloj linearne transformacije. Pojedine težine uče se na različitim dijelovima ulaza i to sve omogućuje veću efikasnost i bolju generalizaciju.

5.4.1. Konvolucija

Konvolucija funkcija iz $\mathbb{Z} \rightarrow \mathbb{R}$ definirana je ovim izrazom:

$$(f * g)(t) := \sum_{\tau} f(\tau)g(t - \tau). \quad (5.22)$$

Jednako tako, definirana je konvolucija funkcija iz $\mathbb{Z}^n \rightarrow \mathbb{R}$:

$$(f * g)(\mathbf{t}) := \sum_{\boldsymbol{\tau}} f(\boldsymbol{\tau})g(\mathbf{t} - \boldsymbol{\tau}). \quad (5.23)$$

Na isti način, s integralom umjesto zbroja, definirana je i konvolucija funkcija s kontinuiranom domenom. Neka od svojstava konvolucije su:

1. Komutativnost: $f * g = g * f$.
2. Distributivnost zbrajanja. Vrijedi $(f + g) * h = f * h + g * h$.
3. Translacijska ekvivarijantnost. Ako $f'(t) := f(t + d)$, onda $(f' * g)(t) = (f * g)(t + d)$.
4. Konvolucija u vremenskoj domeni odgovara umnošku u Fourierovoj domeni, tj. $F[f * g] = F[f]F[g]$, gdje F označava odgovarajuću Fourierovu transformaciju.

Konvolucija se može poopćiti na funkcije s kodomenom koja može općeniti vektorski prostor, tj. funkcije iz $\mathbb{Z}^m \rightarrow \mathbb{R}^n$. Jedan način je ovaj, gdje se po svakoj komponenti paralelno obavlja konvolucija:

$$(f *_{\mathbf{p}} g)(\mathbf{t}) := \sum_{\boldsymbol{\tau}} f(\boldsymbol{\tau}) \odot g(\mathbf{t} - \boldsymbol{\tau}). \quad (5.24)$$

Drugi način je ovaj, gdje se izlazni vektori funkcija skalarno množe:

$$(f *_{\mathbf{s}} g)(\mathbf{t}) := \sum_{\boldsymbol{\tau}} \langle f(\boldsymbol{\tau}) | g(\mathbf{t} - \boldsymbol{\tau}) \rangle. \quad (5.25)$$

U ovom slučaju, kodomena funkcije $f *_{\mathbf{s}} g$ je \mathbb{R} . Isti izraz vrijedi i ako je kodomena funkcija f i g neki skup n -dimenzionalnih nizova, tj. $\mathbb{R}^{d_1 \times \dots \times d_n}$, gdje su d_i pojedine dimenzije niza. Zato se za skalarni produkt ovdje koristi oznaka skalarnog produkta.

5.4.2. Konvolucijski sloj

Jednom umjetnom neuronu kod konvolucijskih mreža, ako se zanemari pomak, obično odgovara operacija u jednadžbi (5.25), samo što funkcijama f i g odgovaraju konačni $(m + 1)$ -dimenzionalni (ili m -dimenzionalni ako $n = 1$) nizovi, pa treba prilagoditi definiciju konvolucije za nizove. Jednoj funkciji odgovara ulazni niz, a drugoj **konvolucijska jezgra (filtar)** koja je obično manja i neovisna o veličini ulaza. Izlaz konvolucije je onda m -dimenzionalni niz kojem su dimenzije obično iste kao prvih m dimenzija ulaznog niza, ovisno o prilagodbi definicije konvolucije na nizove. Ovakvu konvoluciju ćemo nazivati **m -dimenzionalna konvolucija**. Ovdje se neće razmatrati m -dimenzionalna konvolucija $(m + n)$ -dimenzionalnih nizova kod kojih $n > 1$, tj. $\mathbf{A}_{[i_1, \dots, i_m, :]}$ su vektori ako je \mathbf{A} $(m + 1)$ -dimenzionalan.

Slojevi koji obavljaju konvoluciju nazivaju se **konvolucijski slojevi**. Izlaz jedne

jedinice (dobiven jednim filtrom) u konvolucijskom sloju naziva se **mapa značajki**. Izlaz konvolucijskog sloja sastoji se od više mapa značajki i čini $(m + 1)$ -dimenzionalni izlaz kojem je zadnja dimenzija jednaka broju mapa značajki. m -dimenzionalnu konvoluciju s k jezgri nazivat ćemo **k -struka m -dimenzionalna konvolucija**.

Osnovni način definiranja m -dimenzionalne konvolucije (unakrsne korelacije ako ne reflektiramo jezgru) $(m + 1)$ -dimenzionalnog ulaza \mathbf{X} s $(m + 1)$ -dimenzionalnom jezgrom \mathbf{W} , što daje m -dimenzionalni niz $\mathbf{X} *_s \mathbf{W}$, može se ovako izraziti:

$$(\mathbf{X} *_s^v \mathbf{W})_{[t]} := \left\langle \mathbf{X}_{[t:t+d_W+1,:]} \middle| \mathbf{W} \right\rangle, \quad (5.26)$$

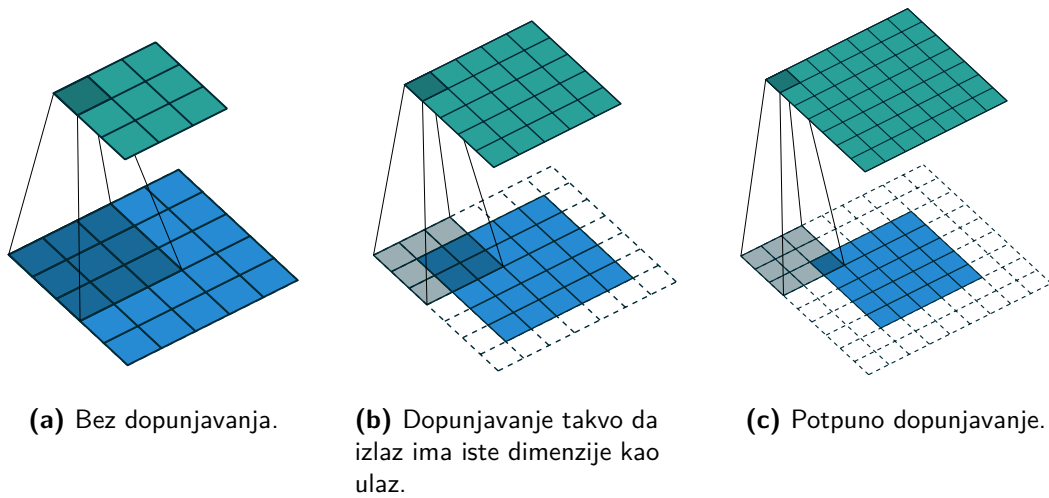
gdje je $\mathbf{d}_W = \dim(\mathbf{W})_{[1:m]}$ vektor dimenzija jezgre po kojima se obavlja konvolucija. Skalarni produkt na desnoj je definiran ako $\forall i \in \{1..m\} \ t_{[i]} \in \{1, \dots, \mathbf{d}_X[i] - \mathbf{d}_W[i] - 1\}$, gdje je $\mathbf{d}_X = \dim(\mathbf{X})_{[1:m]}$ vektor dimenzija ulaza. Izlaz takve operacije je dimenzija $(\mathbf{d}_X[i] - \mathbf{d}_W[i] - 1)$. Kod obrade slike obično želimo da izlaz konvolucije bude jednakih dimenzija kao ulaz. To se može ostvariti dopunjavanjem ulaza nulama po rubu dimenzija po kojima treba obavljati konvoluciju tako da sredina jezgre, za koju pretpostavljamo da ima neparne dimenzije, može doći do ruba izvornog ulaza. Neka $\text{pad}(\mathbf{X}, \frac{1}{2}(\mathbf{d}_W - 1))$ označava takvu operaciju dopunjavanja. Definiramo novu operaciju:

$$\mathbf{X} *_s^s \mathbf{W} := \text{pad}(\mathbf{X}, \frac{1}{2}(\mathbf{d}_W - 1)) *_s^v \mathbf{W}. \quad (5.27)$$

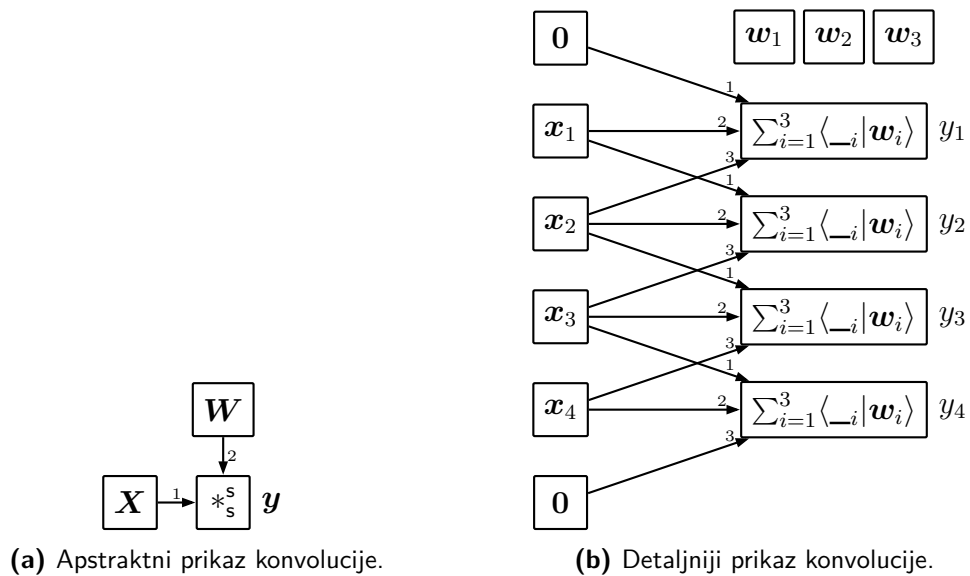
U gornjem indeksu operatora "v" dolazi od riječi *valid* zato što se filter pomiče samo unutar granica ulaza, a "s" od riječi *same* zato što je izlaz istih dimenzija kao ulaz (osim zadnje). Na slici 5.8 ilustrirani su najčešći načina dopunjavanja na primjeru jednostruke dvodimenzionalne konvolucije dvodimenzionalnih nizova. Na slici 5.9 prikazana je jednostruka jednodimenzionalna konvolucija (unakrsna korelacija) dvodimenzionalnih nizova s dopunjavanjem kao u jednadžbi (5.27).

Izlazni korak konvolucije i dilatacija jezgre

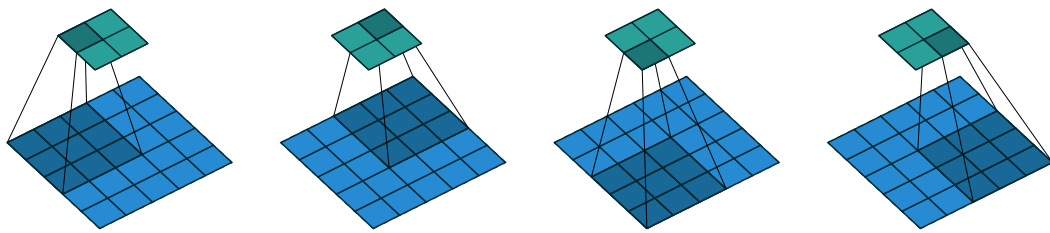
Kod konvolucijski mreža još se koriste neke izmjene konvolucije kako bi se postigla veća računalna efikasnost. Jedna je korištenje **izlaznog koraka** (ili **korak**). Izlazni korak veći od 1 da jezgra po toj dimenziji preskače neke položaje. Na taj način se postiže da dimenzije izlaza budu manje za otprilike za faktor veličine izlaznog koraka. Konvolucija s Izlaznim korakom 2 po svim dimenzijama konvolucije



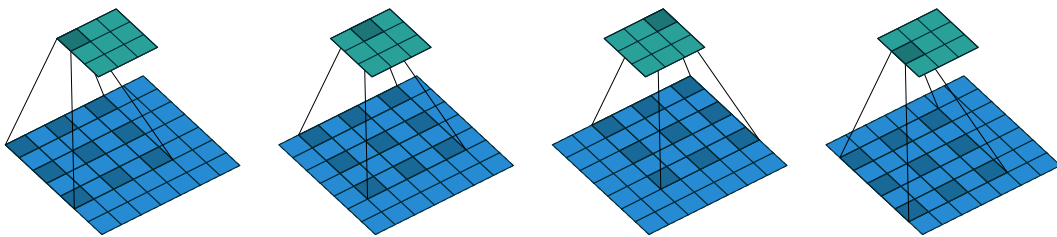
Slika 5.8: Ilustracija dopunjavanja kod dvodimenzionalne konvolucije. Slike 5.8b i 5.8c su preuzete, a slika 5.8a je napravljena na temelju slika iz Dumoulin i Visin (2016).



Slika 5.9: Grafički prikaz jednodimenzionalne konvolucije s dopunjavanjem. Na slici b detaljnije su prikazani dvodimenzionalni nizovi $\mathbf{X} \in \mathbb{R}^{4 \times n}$ i $\mathbf{W} \in \mathbb{R}^{3 \times n}$ iz slike a rastavljeni na vektore, dopunjavanje i konvolucija na razini vektora $\mathbf{x}_i = \mathbf{X}_{[i,:]}$ i $\mathbf{w}_i = \mathbf{W}_{[i,:]}$. Rezultat konvolucije je $\mathbf{y} = [y_1, \dots, y_4] \in \mathbb{R}^4$. $_i$ označava i -ti ulaz čvora u smjeru obrnutom od kazaljke na satu od desne strane.



Slika 5.10: Ilustracija konvolucije s korakom 2. Slike su preuzete iz [Dumoulin i Visin \(2016\)](#).



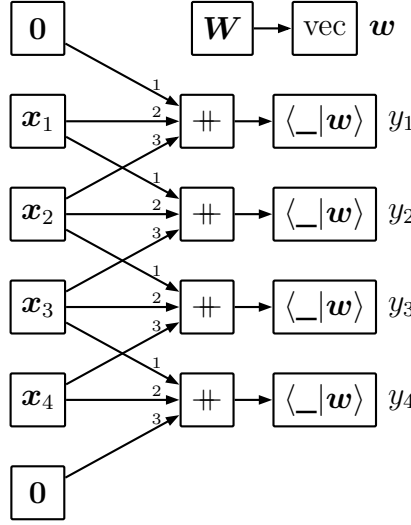
Slika 5.11: Ilustracija konvolucije s dilacijom 1. Slike su preuzete iz [Dumoulin i Visin \(2016\)](#).

ilustrirana ja na slici 5.10.

Kako bi se povećalo **receptivno polje** jedinice konvolucijskog sloja bez povećavanja dimenzija jezgre, koristi se konvolucija s **dilatacijom** (ili **dilacijom**), tj. **širenjem jezgre**. Na slici 5.11 ilustrirana konvolucija s dilacijom 1. Takva konvolucija je ekvivalentan konvoluciji kod koje se koristi veća jezgra kod koje se svaki drugi redak ili stupac sastoji od nula.

Konvolucija kao matrično množenje

Konvolucija je linearna operacija. Na slici 5.12 je konvolucija sa slike 5.9 prikazana malo drugačije. Jezgra je pretvorena u vektor, a ulaz je pretvoren u vektore koji se skalarno množe s vektorom koji predstavlja jezgru. Možemo ulaz \mathbf{X} pretvoriti u matricu $\mathbf{X}_M \in \mathbb{R}^{4 \times 3n}$, a jezgru \mathbf{W} u vektor $w \in \mathbb{R}^{3n}$ tako da njihov



Slika 5.12: Alternativni prikaz konvolucije ekvivalentan onom na slici 5.9. \oplus ovdje označava združivanje vektora $\mathbf{x}_i \in \mathbb{R}^n$ u vektor iz \mathbb{R}^{3n} , vec funkciju koja $\mathbf{W} \in \mathbb{R}^{3 \times n}$ preslikava u $\mathbf{w} \in \mathbb{R}^{3n}$.

matrični umnožak daje izlaz konvolucije:

$$\underbrace{\begin{bmatrix} \mathbf{0}_{1 \times n} & \mathbf{x}_1^\top & \mathbf{x}_2^\top \\ \mathbf{x}_1^\top & \mathbf{x}_2^\top & \mathbf{x}_3^\top \\ \mathbf{x}_2^\top & \mathbf{x}_3^\top & \mathbf{x}_4^\top \\ \mathbf{x}_3^\top & \mathbf{x}_4^\top & \mathbf{0}_{1 \times n} \end{bmatrix}}_{\mathbf{X}_M} \underbrace{\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix}}_{\text{vec}(\mathbf{W})} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\mathbf{y}}. \quad (5.28)$$

Konvolucijski sloj obično ima više jezgri \mathbf{W}_i . Sada se lako vidi da vrijedi $\frac{\partial \mathbf{y}}{\partial \text{vec}(\mathbf{W})} = \mathbf{X}_M$. To možemo poopćiti na k -struku konvoluciju:

$$\mathbf{X}_M \begin{bmatrix} \text{vec}(\mathbf{W}_1) & \text{vec}(\mathbf{W}_2) & \cdots & \text{vec}(\mathbf{W}_k) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_k \end{bmatrix}. \quad (5.29)$$

To se može poopćiti i na višedimenzionalnu konvoluciju (Chetlur et al., 2014). Onda su reci matrice \mathbf{X}_M vektori $\text{vec}\left(\text{pad}\left(\mathbf{X}, \frac{1}{2}(\mathbf{d}_W - \mathbf{1})\right)_{[t:t+\mathbf{d}_W[1:m]+1, :]}\right)$ redom po t , uz oznake iz jednadžbe (5.26), tj. reci su vektori koji sadrže elemente ulaza koje pokriva jezgra za svaki položaj. Jezgra je opet vektor, a kao izlaz se dobije vektor koji treba preoblikovati tako da mu prvih m dimenzija bude jednako prvih m dimenzija ulaza.

Drugi način pretvaranja konvolucije u matrično množenje je ovakav:

$$\underbrace{\begin{bmatrix} \mathbf{w}_2^\top & \mathbf{w}_3^\top & \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times n} \\ \mathbf{w}_1^\top & \mathbf{w}_2^\top & \mathbf{w}_3^\top & \mathbf{0}_{1 \times n} \\ \mathbf{0}_{1 \times n} & \mathbf{w}_1^\top & \mathbf{w}_2^\top & \mathbf{w}_3^\top \\ \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times n} & \mathbf{w}_1^\top & \mathbf{w}_2^\top \end{bmatrix}}_{\mathbf{W}_M} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix}}_{\text{vec}(\mathbf{X})} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\mathbf{y}}. \quad (5.30)$$

Ovdje se vidi da $\frac{\partial \mathbf{y}}{\partial \text{vec}(\mathbf{X})} = \mathbf{W}_M$. Gradijent gubitka L po $\text{vec}(\mathbf{X})$ je $\left(\frac{\partial L}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \text{vec}(\mathbf{X})}\right)^\top = \mathbf{W}_M^\top \nabla_{\mathbf{y}} L$. To isto odgovara jednoj vrsti konvolucije koja se naziva **transponirana konvolucija** (Šegvić, 2018).

5.4.3. Slojevi sažimanja

U konvolucijskim mrežama se, uglavnom radi smanjivanja dimenzija, mogu koristiti **slojevi sažimanja**. Operacije sažimanja, slično konvolucijskim slojevima, primjenjuju neku funkciju pomicanjem okna po dimenzijama konvolucije, obično s korakom većim od 1. Za razliku od konvolucijskih slojeva, oni obično djeluju na svakoj mapi značajki posebno i izlazi sažimanja su invarijantni na zamjenu elemenata unutar okna. To svojstvo se naziva **lokalna invarijantnost**. Najčešće se kao funkcija koja preslikava skup elementa okna u izlaz koristi \max ili prosjek. Veličina okna je često jednaka veličini koraka tako da se susjedna okna ne preklapaju. Na slici 5.13 ilustrirani su primjeri sažimanja.

Za smanjivanje mapa značajki često se koriste i uobičajeni postupci interpolacije¹ slika kao što su **interpolacijski postupak najbližeg susjeda** i **bilinearna interpolacija**.

¹https://en.wikipedia.org/wiki/Multivariate_interpolation

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

(a) Sažimanje prosječnom vrijednošću s oknom dimenzija 3×3 i korakom 1.

(b) Sažimanje maksimalnom vrijednošću s oknom dimenzija 3×3 i korakom 1.

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0
3.0	2.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0

(c) Sažimanje maksimalnom vrijednošću s oknom dimenzija 2×2 i korakom 2.

(d) Globalno sažimanje maksimalnom vrijednošću.

Slika 5.13: Ilustracije primjera dvodimenzionalnog sažimanja. Slike su preuzete iz Dumoulin i Visin (2016) i prilagođene.

6. Procjena nesigurnosti kod dubokih modela

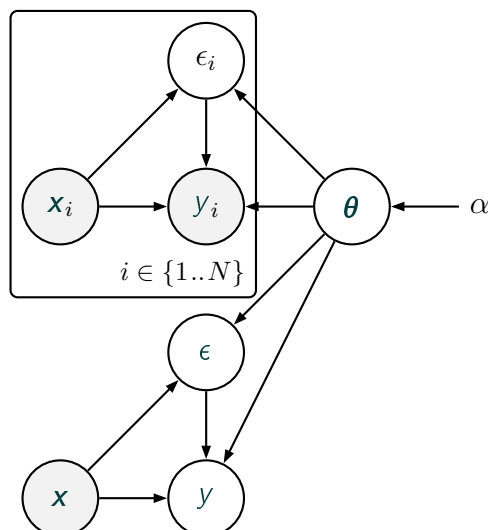
Kod uobičajenih modela dubokog učenja ne možemo pouzdano procijeniti nesigurnost predikcija. Modeli za regresiju kao izlaz daju točkastu procjenu izlaza, a modeli za klasifikaciju daju vektor koji predstavlja razdiobu sigurnosti u klase, ali ta razdioba nije dobar pokazatelj stvarne nesigurnosti i neznanja.

Ovo poglavlje opisuje podjelu nesigurnosti, njenu ulogu i važnost u nekim algoritmima strojnog učenja i neke pristupe koji omogućuju bolje procjenjivanje nesigurnosti kod dubokih nadziranih modela. Rezultati eksperimenata s nekim postupcima opisani su u poglavlju 7.

6.1. Aleatorna i epistemička nesigurnost

Postoje različiti izvori nesigurnosti (C. Kennedy i O'Hagan, 2002), ali nesigurnost općenito možemo podijeliti na dvije vrste: **aleatornu nesigurnost** i **epistemičku nesigurnost** (Kiureghian i Ditlevsen, 2009). Riječ *aleatorna* izvedena je vjerojatno od latinske riječi *aleator* (Gal, 2016) koja znači *kockar*, a riječ *epistemička* izvedena je od grčke riječi *epistēmē* koja znači *znanje*. Aleatorna nesigurnost je nesigurnost koju model ne može smanjiti neovisno o znanju i količini dostupnih podataka. Ona dolazi od višeznačnosti podatka, tj. nedeterminizma samog procesa koji generira podatke, nedostupnosti dijela informacija ili ograničenja modela. Epistemička nesigurnost je nesigurnost u **strukturu modela** i **parametre modela** (Gal, 2016). Ona se zato još naziva **nesigurnost modela**. Ona dolazi od neznanja i može se smanjiti uz više informacija koje mogu biti veći skup podataka za učenje ili induktivna pristranost.

Razlikovanje aleatorne i epistemičke nesigurnosti ovisi o modelu. Nešto što je kod

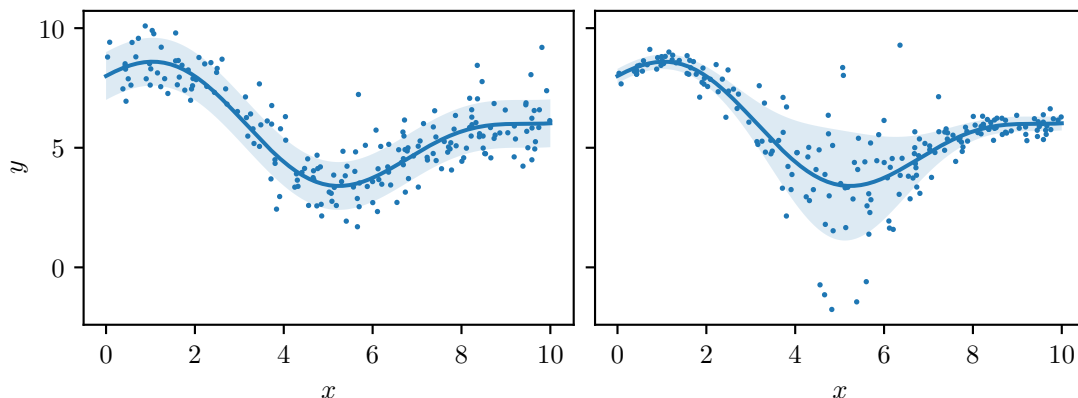


Slika 6.1: Model regresije (prema onom na slici 3.3) kod kojeg su θ nepoznati parametri, x opaženi ulaz, y nepoznati izlaz, a ϵ heteroskedastički šum koji ovisi o ulazu x . Čvorovi s indeksima i predstavljaju podatke za učenje (opaženi čvorovi) i odgovarajući šum (ϵ_i).

jednostavnijeg modela aleatorna nesigurnost, kod složenijeg modela može biti epistemička, tj. može se smanjiti uz više podataka. Ako su neke pojave po prirodi nasumične ili se ne mogu ili ne žele modelu dati informacije koje bi ih mogle objasniti, nesigurnost zaključivanja u vezi tih pojava će biti aleatorna neovisno o ograničenosti modela.

Na temelju aleatorne i epistemičke nesigurnosti može se procijeniti **nesigurnost predikcije**. Kod bayesovskih modela nesigurnost predikcije izražava se razdiobom po vrijednostima varijable čija vrijednost se procjenjuje, a može se izraziti i nekom mjerom kao što je entropija ili varijanca, ovisno o tome što je prikladno.

Aleatorna nesigurnost može biti **homoskedastička** ili **heteroskedastička**. Kažemo da je aleatorna nesigurnost homoskedastička ako je neovisna o primjeru, a heteroskedastička ako ovisi o primjeru. Heteroskedastička nesigurnost se treba modelirati kao funkcija primjera. Za izlaz te funkcije se isto može procjenjivati epistemička nesigurnost — nesigurnost u procjenu aleatorne nesigurnosti. Na slici 6.1 je prikazan primjer grafičkog modela koji pretpostavlja aleatorni šum, a na slici 6.2 je ilustrirana usporedba regresijskih zadataka bez i sa šumom koji ovisi u ulaznom primjeru. Modeliranje aleatorne nesigurnosti se ostvaruje kao funkcija ulaza.



Slika 6.2: Homoskedastički (lijevo) i heteroskedastički (desno) Gaussov šum. Crta prikazuje očekivanje $f(x)$, svjetloplava površina standardnu devijaciju šuma $s(x)$, a točke slučajne uzorke. Točke su generirane prema $(y | x) \sim \mathcal{N}(f(x), s(x)^2)$. Na lijevoj slici je $s(x) = 1$.

6.1.1. Izvanrazdiobni primjeri

Jedan poseban slučaj epistemičke nesigurnosti je nesigurnost u to **pripada li primjer razdiobi skupa za učenje**. Modelu se dati ulazni primjer za koji ne postoji točna oznaka i nije sličan primjerima u skupu za učenje. Takav primjer može biti npr. slika koja pripada nekoj klasi koja nije među onima koje model treba raspoznavati ili samo slučajni šum. Primjeri koji su izvan razdiobe skupa za učenje dalje ćemo nazivati **izvanrazdiobni primjeri**. Primjeri koji su iz razdiobe skupa za učenje nazivat ćemo **unutarrazdiobni primjeri**.

Problem prepoznavanja primjera koji su izvan razdiobe skupa za učenje prirodno rješavaju generativni probabilistički modeli, ali kod složenih visokodimenzionalnih podataka to postaje problem zbog prokletstva dimenzionalnosti i složenosti modeliranja i statističkog zaključivanja. Kod diskriminativnih modela je problem to što oni ne modeliraju razdiobu ulaznih primjera $p(\mathbf{x})$, nego samo uvjetnu razdiobu izlaza uz dani ulaz $p(\mathbf{y} | \mathbf{x})$.

6.2. Važnost i primjene procjene i razlikovanja nesigurnosti

Sve se više upotrebljavaju složeni duboki modeli kod ozbiljnih primjena kao što su medicina i autonomna vozila, gdje treba osigurati pouzdanost i sposobnost prepoznavanja primjera o kojima model ne može donositi pouzdane zaključke.

Trenutno najuspješniji modeli imaju problema s pokazivanjem prevelike sigurnosti u predikciji kod izvanrazdiobnih primjera i krivo klasificiranih primjera [Nguyen et al. \(2015\)](#); [Guo et al. \(2017\)](#); [Hendrycks i Gimpel \(2016\)](#), a mogu se i pronalaziti neprijateljski primjeri ([Szegedy et al., 2013](#); [Goodfellow et al., 2014](#); [Moosavi-Dezfooli et al., 2016](#)), koji su opisani u pododjeljku 5.3.6.

Primjeri područja strojnog učenja u kojima je posebno važna procjena nesigurnosti su aktivno učenje i podržano učenje. Aktivno učenje je oblik polunadziranog učenja gdje algoritam učenja bira primjere za koje zaključi da su potrebni za učenje. Kod podržanog učenja algoritam bira koja će stanja istraživati. Osim procjene mjere nesigurnosti, za efikasno učenje kod takvih algoritama bitno je i razlikovanje epsitemičke i aleatorne nesigurnosti ([Gal, 2016](#)). Npr. agent kod podržanog učenja neće puno naučiti pretraživanjem stanja koja su sama po sebi nasumična (aleatorna), dok će pretraživanjem stanja o kojima nema puno znanja (epistemička nesigurnost) naučiti nešto novo. Dakle, uspješnim modeliranjem i razlikovanjem nesigurnosti može se poboljšati efikasnost takvih algoritama.

6.3. Bayesovske neuronske mreže

Dijelovi ovog odjeljka se temelje na [Gal \(2016, poglavlje 2\)](#).

Bayesovske neuronske mreže su predložene otprilike početkom 90-ih godina prošlog stoljeća ([Denker i LeCun, 1990](#); [MacKay, 1992b](#); [Neal, 1995](#)). Za razliku od običnih neuronskih mreža, gdje se učenjem provodi točkasta procjena parametara, kod bayesovskih neuronskih mreža se provodi bayesovska procjena parametara (pododjeljak 3.2.5), tj. zaključuje se o aposteriornoj razdiobi parametara na temelju apriorne razdiobe i podataka. Tako naučena mreža predstavlja razdiobu nad hipotezama, što omogućuje otpornost na prenaučenosť i procjenjivanje nesigurnosti predikcija ([Gal, 2016](#)).

Kod bayesovskih neuronskih mreža se za apriornu razdiobu težina često koristi pretpostavka nezavisnosti i Gaussova razdioba $\mathcal{N}(0, \lambda^{-1})$ za svaku težinu, gdje je λ preciznost. Radi jednostavnosti se pomaci često točkasto procjenjuju. Iako su bayesovske neuronske mreže jednostavne za formulirati, kod njih nije jednostavno provoditi zaključivanje.

Prema jednadžbi (3.18), aposteriorna vjerojatnost parametara je

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})}, \quad (6.1)$$

gdje je

$$p(\mathcal{D}) = \int p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \mathbf{E}_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) \quad (6.2)$$

marginalna izglednost koja se računa marginalizacijom nazivnika po parametrima. Ta marginalizacija ovdje predstavlja glavni problem i aposteriorna razdioba se mora aproksimirati postupcima kao što su oni navedeni u odjeljku 3.4. Na temelju ulaza i aposteriorne razdiobe parametara provodi se zaključivanje o izlazu (isto kao jednadžba (6.3)):

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} = \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}), \quad (6.3)$$

Dvije osnovne skupine pristupa aproksimaciji aposteriorne razdiobe su kratko opisane u odjeljku 3.4. Kod jedne skupine (MCMC) definira se Markovljev lanac kod kojeg je stacionarna razdioba jednaka aposteriornoj razdiobi parametara i simulacijom se dobivaju uzorci parametara koji se mogu koristiti u *Monte Carlo* aproksimacijama (npr. desnih strana jednadžbi (6.2) i (6.3)). Druga skupina je varijacijsko zaključivanje (opisano u odjeljku 3.5), gdje se aposteriorna razdioba parametara zamijeni nekom jednostavnijom razdiobom za koju se optimizacijski traže parametri koji ju najviše približavaju aposteriornoj razdiobi.

6.3.1. Varijacijsko zaključivanje kod bayesovskih neuronskih mreža

Kod varijacijskog zaključivanja, koje je opisano u odjeljku 3.5, za bayesovske neuronske mreže tražimo varijacijsku razdiobu koja minimizira KL-divergenciju prema jednadžbi (3.27):

$$q^* = \arg \min_{q_{\phi}} D_{\text{KL}}(q_{\phi} \parallel p(\boldsymbol{\theta} \mid \mathcal{D})), \quad (6.4)$$

Varijacijske razdiobe koje su definirane varijacijskim parametrima ϕ ovdje su predstavljene slučajnim varijablama $\tilde{\boldsymbol{\theta}}$. Minimizacija s obzirom na parametre varijacijske razdiobe je ekvivalentna maksimizaciji donje granice logaritma

marginalne izglednosti

$$L_{\mathbb{D}}(\tilde{\boldsymbol{\theta}}) = \mathbf{E}_{\tilde{\boldsymbol{\theta}} \sim q_{\phi}} \ln p(\mathbb{D} \mid \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) - D_{\text{KL}}(q_{\phi} \parallel p(\boldsymbol{\theta})), \quad (6.5)$$

za koju ne treba računati marginalnu izglednost $p(\mathbb{D})$. Zbog pretpostavke nezavisnosti primjera i pretpostavke diskriminativnog modela $\mathbf{x} \perp \boldsymbol{\theta}$ vrijedi

$$p(\mathbb{D} \mid \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) = \prod_i p(\mathbf{x}_i, \mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) = \prod_i (p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) p(\mathbf{x}_i)). \quad (6.6)$$

Možemo zanemariti faktore $p(\mathbf{x}_i)$ jer oni ne ovise o parametrima i maksimiziramo

$$\mathbf{E}_{\tilde{\boldsymbol{\theta}} \sim q_{\phi}} \left(\sum_i \ln p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}) \right) - D_{\text{KL}}(q_{\phi} \parallel p(\boldsymbol{\theta})) \quad (6.7)$$

s obzirom na varijacijske parametre. Prvi član u tom izrazu potiče maksimizaciju izglednosti parametara na skupu za učenje. Razlika u odnosu na maksimizaciju izglednosti kod običnih dubokih mreža je da se izglednost ne maksimizira po točkastim procjenama parametra, nego po razdiobama. Drugi dio izraza kažnjava udaljavanje od apriorne razdiobe i služi kao regularizacija.

Zamjenom aposteriorne razdiobe u jednadžbi (6.3) sa zamjenskom razdiobom q_{ϕ} , zaključivanje o izlazu mreže postaje

$$p(\mathbf{y} \mid \mathbf{x}, \mathbb{D}) \approx \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) q_{\phi}(\boldsymbol{\theta}) d\boldsymbol{\theta} = \mathbf{E}_{\tilde{\boldsymbol{\theta}} \sim q_{\phi}} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}). \quad (6.8)$$

U pododjeljku 6.6.1 je opisana aproksimacija bayesovske neuronske mreže pomoću *dropout*.

6.4. Mjere za izražavanje nesigurnosti predikcije

Osnovne mjere za izražavanje nesigurnosti su vjerojatnost, entropija i varijanca. Pretpostavimo da modeliramo (diskretnu ili kontinuiranu) razdiobu $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ ili $p(\mathbf{y} \mid \mathbf{x}, \mathbb{D})$ ako se provodi bayesovsko zaključivanje. Jedan način izražavanja nesigurnosti u predikciju kod klasifikacije je vjerojatnost klase s najvećom vjerojatnošću

$$\max_k P(\mathbf{y} = k \mid \mathbf{x}, \boldsymbol{\theta}). \quad (6.9)$$

Nesigurnost se može izraziti i entropijom

$$H(y | \mathbf{x}, \boldsymbol{\theta}) = - \mathbf{E}_{y|\mathbf{x},\boldsymbol{\theta}} \ln P(y | \mathbf{x}, \boldsymbol{\theta}) \quad (6.10)$$

ili, kod regresije, diferencijalnom entropijom

$$h(y | \mathbf{x}, \boldsymbol{\theta}) = - \mathbf{E}_{y|\mathbf{x},\boldsymbol{\theta}} \ln p(y | \mathbf{x}, \boldsymbol{\theta}) \quad (6.11)$$

koja je, ako $(y | \mathbf{x}, \boldsymbol{\theta})$ ima Gaussovu razdiobu, proporcionalna (i kao mjera ekvivalentna) varijanci $\mathbf{D}(y | \mathbf{x}, \boldsymbol{\theta})$. Ako kod klasifikacije gledamo samo vjerojatnost klase s najvećom vjerojatnošću, možemo kao mjeru nesigurnosti koristiti i binarnu entropiju $-p_{\hat{y}} \ln p_{\hat{y}} - (1 - p_{\hat{y}}) \ln(1 - p_{\hat{y}})$, gdje je $p_{\hat{y}}$ vjerojatnost klase s najvećom vjerojatnošću.

Jedan nedostatak ovih mjera je što ne razlikuju epistemičku i aleatornu nesigurnost. Još jedan mogući nedostatak entropije i maksimalne vjerojatnosti može biti to što npr. razdioba $(0.6, 0.2, 0.2)$ ima veću entropiju od razdiobe $(0.6, 0.4, 0)$ iako se može interpretirati da je kod prve razdiobe veća jednoznačnost zaključka o klasi.

6.5. Razlikovanje aleatorne i epistemičke nesigurnosti

6.5.1. Eksplicitno modeliranje aleatorne nesigurnosti

Kendall i Gal (2017) za bayesovske neuronske mreže predlažu eksplicitno modeliranje aleatorne nesigurnosti predikcijom varijance kod regresije i varijance logita kod klasifikacije. Kao konačnu procjenu nesigurnosti, za regresiju procjenjuju ukupnu varijancu izlaza koja je zbroj procjene aleatorne varijance i epistemičke varijance, a za klasifikaciju entropiju izlaza koji se dobiva marginalizacijom po parametrima i logitima za koje pretpostavljaju faktoriziranu Gaussovu razdiobu. Za regresiju predlažu gubitak proporcionalan negativnom logaritmu izglednosti koji uključuje varijancu, koja se inače ne uključuje u gubitak jer se pretpostavlja da je konstanta. Za regresiju je to

$$L(\mathbf{y} | h'(\mathbf{x}; \boldsymbol{\theta})) = \frac{1}{2\sigma(\mathbf{x}; \boldsymbol{\theta})^2} \|\mathbf{y} - h(\mathbf{x}; \boldsymbol{\theta})\|^2 + \frac{1}{2} \ln \sigma(\mathbf{x}; \boldsymbol{\theta})^2, \quad (6.12)$$

gdje $h'(\mathbf{x}; \boldsymbol{\theta}) = (h(\mathbf{x}; \boldsymbol{\theta}), \sigma(\mathbf{x}; \boldsymbol{\theta}))$. U slučaju homoskedastičkog šuma $\sigma(\mathbf{x}; \boldsymbol{\theta}) = \sigma$, gdje je σ parametar modela. Takav gubitak potiče predikciju veće varijance kada hipoteza očekuje veću kvadratnu pogrešku. Radi numeričke stabilnosti, [Kendall i Gal \(2017\)](#) predlažu da se umjesto $\sigma(\mathbf{x}; \boldsymbol{\theta})$ kao predikcija daje $s(\mathbf{x}, \boldsymbol{\theta}) = \ln \sigma(\mathbf{x}; \boldsymbol{\theta})^2$. Gubitak je onda

$$L(\mathbf{y} | h'(\mathbf{x}; \boldsymbol{\theta})) = \frac{1}{2} \exp(-s(\mathbf{x}; \boldsymbol{\theta})) \|\mathbf{y} - h(\mathbf{x}; \boldsymbol{\theta})\|^2 + \frac{1}{2} s(\mathbf{x}; \boldsymbol{\theta}). \quad (6.13)$$

Za procjenu ukupne varijance predikcije predlažu zbroj aleatorne varijance i očekivanja varijanci elemenata izlaza:

$$\mathbf{E}_{\boldsymbol{\theta} | \mathbb{D}} \sigma(\mathbf{x}; \boldsymbol{\theta}) + \sum_i \mathbf{D}_{\boldsymbol{\theta} | \mathbb{D}} (h(\mathbf{x}; \boldsymbol{\theta})_{[i]}). \quad (6.14)$$

Za procjenu epistemičke nesigurnosti predikcije kod klasifikacije [Kendall i Gal \(2017\)](#) koriste entropiju očekivanja predikcije: $H(\mathbf{y} | \mathbf{x}, \mathbb{D}) = H(\mathbf{E}_{\boldsymbol{\theta} | \mathbb{D}}(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}))$. Razdiobe $(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta})$ predstavlja izlaz softmaxa, tj. $h(\mathbf{x}; \boldsymbol{\theta})_{[k]} = (y = k | \mathbf{x}, \boldsymbol{\theta})$. Za procjenu aleatorne nesigurnosti kod klasifikacije pretpostavljaju da logiti imaju Gaussovu razdiobu i predlažu da se za svaki logit procjenjuje varijanca. Neka je $g(\mathbf{x}) = g(\mathbf{x}; \boldsymbol{\theta})$ funkcija koja daje očekivanje, a $\sigma(\mathbf{x}) = \sigma(\mathbf{x}; \boldsymbol{\theta})$ funkcija koja daje vektor standardnih devijacija logita. Izlazna razdioba se računa ovako:

$$h(\mathbf{x}; \mathbb{D}) = \mathbf{E}_{s \sim \mathcal{N}(g(\mathbf{x}), \text{diag}(\sigma(\mathbf{x})^2))} \text{softmax}(\mathbf{s}). \quad (6.15)$$

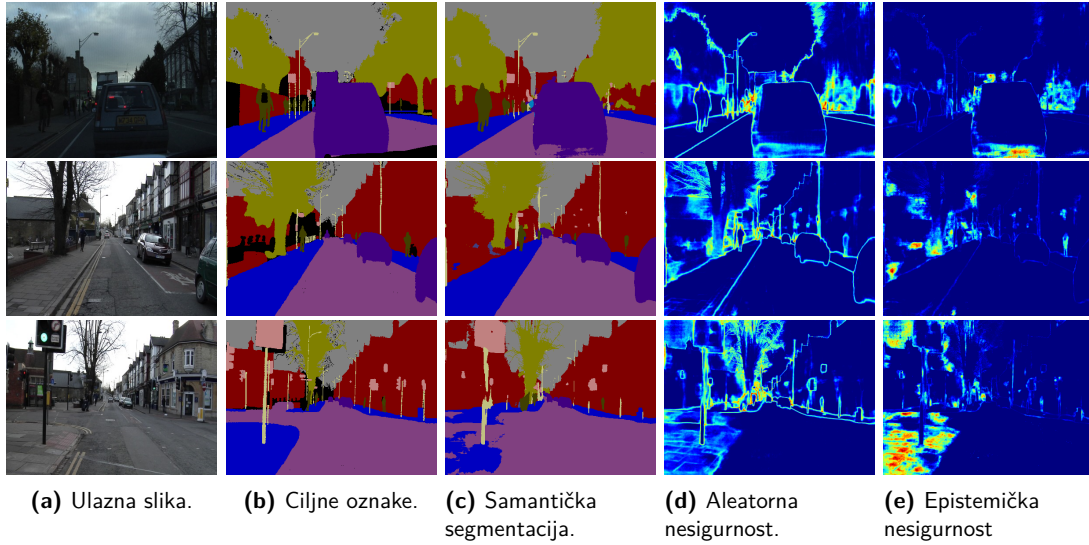
Gubitak ostaje negativni logaritam izglednosti i može se ovako izraziti:

$$L(\mathbf{y} | h'(\mathbf{x}; \boldsymbol{\theta})) = -\ln h(\mathbf{x}; \mathbb{D}) \quad (6.16)$$

$$= \ln \left(\mathbf{E}_{s \sim \mathcal{N}(g(\mathbf{x}), \text{diag}(\sigma(\mathbf{x})^2))} \exp(\mathbf{s} - \ln(\mathbf{1}^\top \exp(\mathbf{s}))) \right). \quad (6.17)$$

Za procjenu ovog i drugih očekivanja (i varijanci) se koristi *Monte Carlo* aproksimacija.

[Kendall i Gal \(2017\)](#) su koristili aproksimaciju bayesovske neuronske mreže pomoću *dropout* i empirijski su pokazali da se uz modeliranje aleatorne nesigurnosti uz gubitke u jednadžbama (6.12) i (6.13) može dobiti malo poboljšanje predikcija na zadacima regresije dubine i semantičke segmentacije. Aproksimaciju bayesovske neuronske mreže pomoću *dropout* je opisana u pododjeljku 6.6.1. Na slici 6.3 se mogu vidjeti primjeri razlikovanja aleatorne i epistemičke nesigurnosti kod predikcije, što su dobili takvim postupkom za semantičku segmentaciju.



Slika 6.3: Ilustracija procjena aleatorne i epistemičke kod semantičke segmentacije preuzeta iz [Kendall i Gal \(2017\)](#). Aleatorna nesigurnost je veća na rubovima objekata, posebno na rubovima krošanja, i na udaljenim objektima.

6.5.2. Međusobna informacija kao mjera epistemičke nesigurnosti

[Rawat et al. \(2017\)](#); [Smith i Gal \(2018\)](#), s ciljem prepoznavanja neprijateljskih primjera, predlažu korištenje međusobne informacije (jednadžbe (2.57)-(2.61)) kod bayesovskog zaključivanja kako bi se razlikovale epistemička i aleatorna nesigurnost. Prema [Smith i Gal \(2018\)](#), količina informacije koju dobijemo o parametrima ako dobijemo oznaku za novi ulazni primjer \mathbf{x} je

$$I(y, \boldsymbol{\theta} \mid \mathbf{x}, \mathcal{D}) = H(y \mid \mathbf{x}, \mathcal{D}) - H((y \mid \mathbf{x}, \mathcal{D}) \mid (\boldsymbol{\theta} \mid \mathbf{x}, \mathcal{D})) \quad (6.18)$$

$$= H(y \mid \mathbf{x}, \mathcal{D}) - \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} H(y \mid \boldsymbol{\theta}, \mathbf{x}, \mathcal{D}) \quad (6.19)$$

$$= H(y \mid \mathbf{x}, \mathcal{D}) - \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} H(y \mid \boldsymbol{\theta}, \mathbf{x}), \quad (6.20)$$

gdje je korištena definicija uvjetne entropije i pretpostavka $y \perp \boldsymbol{\theta} \mid \mathcal{D}$, tj. znanje o podacima ne nosi nove informacije o oznaci ako znamo parametre. Međusobna informacija se može interpretirati kao količina znanja koja se dobije o parametrima kada se opaža oznaka y za primjer \mathbf{x} . Ako nema epistemičke nesigurnosti, tj. nesigurnosti u parametre, izlazne razdiobe $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ će biti jednake. Desni član u izrazu (6.20) će biti jednak lijevom i međusobna informacija će biti 0. Ako nema aleatorne nesigurnost, izlazna razdioba će za svaki parametar $\boldsymbol{\theta}$ iz aposteriorne razdiobe parametara za neki y imati vjerojatnost 1, tj. entropiju će im biti 0. Desni

član u izrazu (6.20) će onda biti 0, a lijevi može biti veći od 0 jer je on entropija prosječne hipoteze za ulaz \mathbf{x} , predikcije različitih parametara iz aposteriorne razdiobe mogu se razlikovati.

Kao što je i ilustrirano na slici 2.4, vrijedi $0 \leq I(y, \boldsymbol{\theta} \mid \mathbf{x}, \mathcal{D}) \leq H(y \mid \mathbf{x}, \mathcal{D})$. Oduzimanjem međusobne informacije od entropije marginalizirane izlazne razdiobe možemo dobiti mjeru aleatorne nesigurnosti:

$$\mathbb{E}_{\boldsymbol{\theta} \mid \mathcal{D}} H(y \mid \boldsymbol{\theta}, \mathbf{x}). \quad (6.21)$$

Smith i Gal (2018) objašnjavaju uspješnost korištenja varijance softmaxa kao mjere epsitemičke nesigurnosti u drugim radovima pokazujući da se ona može dobiti kao aproksimacija međusobne informacije razvojem međusobne informacije u Taylorov red.

Smith i Gal (2018) su koristili aproksimaciju bayesovske neuronske mreže pomoću dropouta, što je opisano u pododjeljku 6.6.1, i koristili su uzorkovanje, tj. *Monte Carlo* aproksimaciju za procjenu izlazne razdiobe i mjera nesigurnosti.

6.6. Primjeri pristupa za procjenu nesigurnosti

U ovom odjeljku su opisani primjeri pristupa za procjenu nesigurnosti i prepoznavanje izvanrazdiobnih i krivo klasificiranih primjera bez puno izmjena uobičajenih dubokih modela. U poglavlju 7 su prikazani rezultati nekih eksperimenata s nekim od opisanih pristupa.

6.6.1. Aproksimacija bayesovske neuronske mreže pomoću dropouta

Dropout, opisan u pododjeljku 5.3.4, je postupak koji tijekom učenja nasumično isključuje jedinice i može se interpretirati kao aproksimacija učenja eksponencijalnog broja mreža koje dijele parametre. Pri testiranju se obično usrednjavanje aproksimira tako da se, umjesto isključivanja jedinica, izlazi slojeva usrednje skaliranjem koje odgovara vjerojatnosti neisključivanja. Drugi način usrednjavanja je usrednjavanje izlaza cijele mreže dobivenih uzorkovanjem uz dropout kao pri učenju Srivastava et al. (2014); Gal i Ghahramani (2015a), što se naziva **MC-dropout** (*Monte Carlo*

dropout). Taj način usrednjavanja je ispravniji i daje bolju performansu, ali manje efikasan jer zahtijeva veći broj evaluacija izlaza uz isključivanje jedinica.

Gal i Ghahramani (2016) učenje s *dropoutum* interpretiraju kao varijacijsko zaključivanje s Bernoullijevom razdiobom kod bayesovske neuronske mreže i pokazuju da korištenje *MC-dropouta* pri testiranju daje bolje rezultate ako je mreža učena na manjem skupu za učenje. Ako pretpostavimo da *dropout* dolazi iz slojeva linearne transformacije kojima odgovaraju matrice \mathbf{M}_l , slučajna varijabla koja odgovara varijacijskoj razdiobi matrice težina je

$$\mathbf{W}_l = \text{diag}([z_{l,i}]_{i=1..n_l}) \mathbf{M}_l, \quad (6.22)$$

gdje je l indeks linearnog sloja, n_l broj redaka matrice težina, tj. broj jedinica sloja, \mathbf{M}_l varijacijski parametri, a $z_{l,i}$ nezavisne slučajne varijable s Bernoullijevom razdiobom s očekivanjem p , ako je $1 - p$ vjerojatnost isključivanja. Druge parametre možemo točkasto procjenjivati, što za svaki parametar odgovara razdiobi koja ima samo jednu moguću vrijednost koja je varijacijski parametar.

Prvi član funkcije definirane izrazom (6.7) (očekivanje) možemo aproksimirati *Monte Carlo* aproksimacijom. Ako zamijenimo redosljed očekivanja i zbroja i odaberemo 1 uzorak po svakom očekivanju, on postaje

$$\sum_i \ln p(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i), \quad (6.23)$$

gdje su $\tilde{\boldsymbol{\theta}}_i$ uzorci parametara iz varijacijske razdiobe q_ϕ . Za svaki primjer se uzima jedan uzorak parametara u jednoj iteraciji ili epohi, kao što je uobičajeno kada se koristi *dropout*. Drugi član u izrazu (6.7) (bez minusa) je

$$D_{\text{KL}}(q_\phi \parallel p(\boldsymbol{\theta})) = \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \ln \frac{q_\phi(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta} \quad (6.24)$$

$$= \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \ln q_\phi(\boldsymbol{\theta}) d\boldsymbol{\theta} - \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \ln p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \quad (6.25)$$

Prvi integral u zadnjem izrazu je baskonačan zato što se faktori varijacijske razdiobe sastoje od Diracovih *šiljaka*. Šiljke možemo aproksimirati proizvoljno uskim pravokutnicima širine 2ϵ , npr.

$$p(\mathbf{W}_{l[i,j]} = w) = (1 - p)\delta(w) + p\delta(w - \mathbf{M}_{l[i,j]}) \quad (6.26)$$

$$\approx \frac{1 - p}{2\epsilon} \mathbb{I}[-\epsilon < w < \epsilon] + \frac{p}{2\epsilon} \mathbb{I}[-\epsilon < w - \mathbf{M}_{l[i,j]} < \epsilon]. \quad (6.27)$$

Ako pretpostavimo da neke težine neće postati točno 0, tj. bliže nuli od 2ϵ , prvi

član u izrazu (6.25) (diferencijalna entropija varijacijske razdiobe) onda postaje konačan i neovisan o varijacijskim parametrima i može se zanemariti kod učenja. Za drugi član u izrazu (6.25) ne aproksimiramo varijacijsku razdiobu. Ona je težinski zbroj višedimenzionalnih Diracovih *šiljaka* kojima se *uzorkuje* apriorna razdioba. Ako se koristi Gaussova apriorna razdioba, može se pokazati da je taj član proporcionalan p i zbroju kvadrata težina, što odgovara L^2 regularizaciji. Malo drugačiji i detaljniji izvod može se vidjeti u Gal i Ghahramani (2015a,b), gdje se mreža s *dropoutom* interpretira kao aproksimacija s Gaussovog procesa¹. Vidimo da ako su regularizirane samo matrice težina, maksimizaciji izraza (6.7) odgovara minimizacija ove funkcije pogreške:

$$E(\boldsymbol{\theta}; \mathcal{D}) = - \sum_i \ln p(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i) + \frac{\lambda}{2} \sum_l \|\mathbf{M}_l\|_F^2, \quad (6.28)$$

gdje $\lambda \in \mathbb{R}_{\geq 0}$. To je ista funkcija pogreške koja se inače koristi kod mreže s *dropoutom*.

Za zaključivanje prema izrazu (6.8) može se koristiti *Monte Carlo* aproksimacija (*MC-dropout*):

$$p(\mathbf{y} | \mathbf{x}, \mathcal{D}) \approx \mathbf{E}_{\tilde{\boldsymbol{\theta}} \sim q_\phi} p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}) \approx \frac{1}{M} \sum_{i=1}^M p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i), \quad (6.29)$$

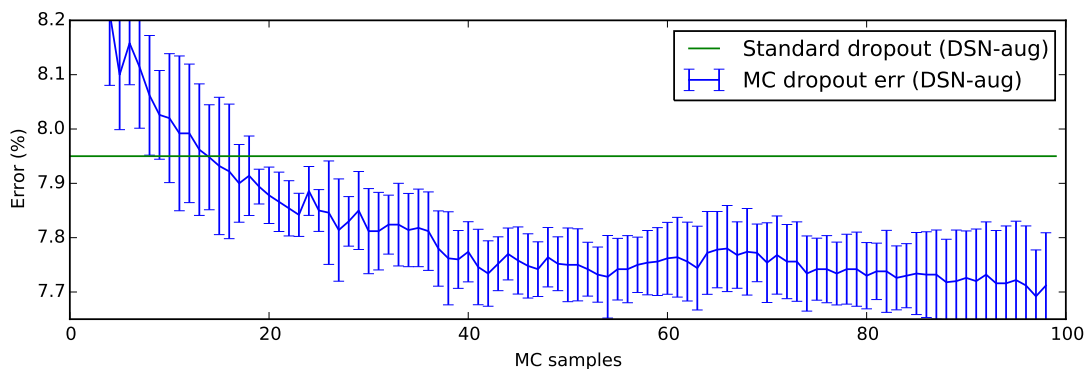
gdje su $\tilde{\boldsymbol{\theta}}_i$ uzorci parametara iz varijacijske razdiobe q_ϕ .

Konvolucijske mreže

Kod konvolucijskih mreža *dropoutom* se obično krši svojstvo ekvivarijantnosti, tj. za svaki položaj jezgre se nezavisno određuje hoće li se element izlaza isključiti. *Dropoutu* kod konvolucijskih slojeva za svaki položaj jezgre odgovara posebna slučajna varijabla prema jednadžbi (6.22), ali sve dijele parametre \mathbf{M}_l . Matrici \mathbf{M}_l kod konvolucijskog sloja odgovara matrica težina kao u jednadžbi (5.29) transponirana tako da jezgre odgovaraju recima.

Slika 6.4 prikazuje ovisnost klasifikacijske pogreške o broju uzoraka *MC-dropouta* na primjeru konvolucijske mreže. Broj uzoraka potrebnih za postizanje bolje performanse ovisi o modelu i skupu podataka. Npr. Srivastava et al. (2014) su za nekonvolucijski model koji su ispitivali na lakšem skupu, MNIST-u, trebali više od 50 uzoraka za postizanje manje klasifikacijske pogreške uz *MC-dropout*.

¹https://en.wikipedia.org/wiki/Gaussian_process



Slika 6.4: Ovisnost klasifikacijske pogreške (plavo) o broju uzoraka u *Monte Carlo* aproksimaciji izlaza na mreži koju su ispitivali autori (Gal i Ghahramani, 2016) na skupu CIFAR-10 (Krizhevsky, 2009). Svaka točka je prosjek 5 mjerenja i prikazane su standardne devijacije. Zeleni pravac označava klasifikacijsku pogrešku kod userdnjavanja kakvo se inače koristi kod testiranja. Slika je preuzeta iz Gal i Ghahramani (2016).

6.6.2. Prepoznavanje izvanrazdiobnih i krivo klasificiranih primjera na temelju izlaza softmaksa ili logita

Kao što je spomenuto u odjeljku 6.2, razdiobe koje duboki modeli daju kao izlaz softmaksa su često previše sigurne kod krive klasifikacije i nije ih dobro interpretirati kao vjerojatnosti.

Hendrycks i Gimpel (2016) pokazuju da se krivo klasificirani i izvanrazdiobni primjeri mogu uspješnije nego što je očekivano prepoznavati klasifikacijom maksimalne vjerojatnosti softmaksa kod različitih zadataka i skupova podataka.

Guo et al. (2017) pokazuju da se samo skaliranjem temperature softmaksa, tj. dodavanjem dijeljenja svih logita (ulaza softmaksa) s T prije softmaksa, može značajno poboljšati kalibracija² već naučene mreže. Kod skaliranja temperature, ako su logiti s ($h(\mathbf{x}) = \text{softmax}(\mathbf{s})$), izlazni vektor vjerojatnosti uz skaliranje temperature je $\text{softmax}(\frac{1}{T}\mathbf{s})$. Optimalni T se traži na validacijskom skupu. Guo et al. (2017) su još isprobali nekoliko složenijih postupaka, ali s njima nisu dobili bolje rezultate. Još nešto zanimljivo što su zaključili je da se optimalne općenitije transformacije logita kao $\text{softmax}(\mathbf{s} \odot \mathbf{t})$ otprilike svode na skaliranje temperature, tj. vrijedi $\mathbf{t}^* \approx T^* \mathbf{1}$ gdje su \mathbf{t}^* i T^* optimalni.

Liang et al. (2017) predlažu dva poboljšanja klasifikacije maksimalne vrijednosti softmaksa. Jedno poboljšanje je skaliranje temperature. Pokazuju da, što je veća

²Model je dobro kalibriran ako svaka vjerojatnost koju djeljuje nekoj klasi slična stvarnoj učestalosti te klase kada model dodjeljuje tu vjerojatnost.

temperatura, to se primjeri izvanrazdiobni primjeri mogu bolje odvojiti od unutarrazdiobnih primjera na temelju maksimalne vrijednosti softmaxa. Drugo poboljšanje je izmjena ulaza mreže tako da se FGSM-om (jednadžba (5.16)) pomakne u smjeru povećavanja maksimalnog izlaza softmaxa (za razliku od smanjivanja kod neprijateljskih primjera):

$$\tilde{\mathbf{x}} = \mathbf{x} - \epsilon \operatorname{sgn} \nabla_{\mathbf{x}} \left(-\ln \left(\max_k h(\mathbf{x})_{[k]} \right) \right). \quad (6.30)$$

ϵ je parametar koji se određuje pomoću izdvojenog skupa izvanrazdiobnih primjera. [Liang et al. \(2017\)](#) pokazuju da takav pomak ulaznog primjera ima veći utjecaj na unutarrazdiobne primjere i tako ih bolje razdvaja od izvanrazdiobnih.

Za ovaj rad su još ispitani slični pristupi kod kojih se umjesto maksimalnog izlaza softmaxa klasificira maskimalni logit

7. Eksperimenti

Eksperimentalno su ispitani neki od pristupa za razlikovanje aleatorne i epistemičke nesigurnosti (odjeljak 6.5) i neki od pristupa za prepoznavanje izvanrazdiobnih primjera (odjeljak 6.6).

Korišten je programski jezik Python i biblioteke TensorFlow, NumPy, PyTorch, Scikit-image, Scikit-learn, Matplotlib, SciPy i druge. Programski kod je u repozitoriju <https://github.com/lvan1248/deep-learning-uncertainty>.

7.1. Evaluacijske mjere za klasifikaciju

U ovom odjeljku su opisane evaluacijske mjere korištene u eksperimentima.

7.1.1. Binarna klasifikacija

Kod binarne klasifikacije klase dijelimo na **pozitivnu klasu** i **negativnu klasu**. Primjere iz skupa korištenog za ispitivanje prema predikcijama klasifikatora dijelimo u 4 skupine:

1. **stvarno pozitivni** (engl. *true positives*, TP) — pozitivno klasificirani pozitivni primjeri
2. **stvarno negativni** (engl. *true negatives*, TN) — negativno klasificirani negativni primjeri
3. **lažno pozitivni** (engl. *false positives*, FP) — pozitivno klasificirani negativni primjeri
4. **lažno negativni** (engl. *true negatives*, FN) — negativno klasificirani pozitivni primjeri.

		y	
		1	0
$h(\mathbf{x})$	1	TP	FP
	0	FN	TN

Slika 7.1: Konfuzijska matrica kod binarne klasifikacije. Stupcima odgovaraju stvarne klase, a recima predikcije.

Neka $h(\mathbf{x}_i)$ označava predikciju modela za primjer \mathbf{x}_i čija je stvarna oznaka y_i .

Ovako ćemo označavati brojeve primjera u navedenim skupinama:

$$\begin{aligned} TP &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = 1] \mathbb{I}[y_i = 1], & FP &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = 1] \mathbb{I}[y_i = 0], \\ FN &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = 0] \mathbb{I}[y_i = 1], & TN &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = 0] \mathbb{I}[y_i = 0]. \end{aligned} \quad (7.1)$$

To možemo prikazati konfuzijskom matricom kao na slici 7.1. Ako je N ukupan broj primjera, vrijedi $N = TP + TN + FP + FN$.

Neke od češćih evaluacijskih mjera za binarnu klasifikaciju su:

1. **točnost** (engl. *accuracy*) — udio točno klasificiranih primjera:

$$A := \frac{TP+TN}{N} = \frac{TP+TN}{TP+FP+TN+FN}$$

2. **preciznost** (engl. *precision*) — udio stvarno pozitivnih primjera u pozitivno klasificiranim primjerima: $P := \frac{TP}{TP+FP}$

3. **odziv** ili **stopa stvarnih pozitivna** (engl. *recall, true positive rate*) — udio stvarno pozitivnih primjera u pozitivnim primjerima: $R := TPR := \frac{TP}{TP+FN}$

4. **stopa lažnih pozitivna** (engl. *false positive rate*) — udio lažno pozitivnih primjera u negativnim primjerima: $FPR := \frac{FP}{TN+FP}$

5. **mjera F_1** - harmonijska sredina preciznosti i odziva:

$$F_1 := \frac{2}{P^{-1}+R^{-1}} = \frac{2PR}{P+R} = \frac{2TP}{2TP+FP+FN}$$

6. **Jaccardov indeks** ili **omjer presjeka i unije** (engl. *intersection over union*) - udio točno klasificiranih primjera u uniji pozitivnih i pozitivno klasificiranih primjera: $J := IoU := \frac{TP}{TP+FP+FN} = \frac{1}{P^{-1}+R^{-1}-1}$.

Kod binarne klasifikacije (ne kod višeklasne) su F_1 i J ekvivalentne u smislu da se jedna može izraziti preko druge — F_1 je harmonijska sredina između J i 1:

$$F_1 = \frac{2}{J^{-1}+1}.$$

Mjere neovisne o pragu klasifikatora

Obično se kod binarne klasifikacije kao izlaz dobiva realni broj koji može, ali ne mora, predstavljati vjerojatnost pozitivne klase. Ako je on veći od praga, primjer se klasificira u pozitivnu klasu, a inače u negativnu. Promjenom praga možemo mijenjati odnos preciznosti i odziva. Za najniži prag je odziv 1, a za najviši je 0. Ovisnost preciznosti o odzivu se može prikazati monotono padajućom krivuljom koju nazivamo **krivulja preciznosti i odziva**, kraće **PR-krivulja**. Klasifikator možemo evaluirati neovisno o pragu računanjem **prosječne preciznosti** koja je definirana kao površina ispod PR-krivulje, tj. integriranjem preciznosti kao funkcije o odzivu:

$$AP := \int_0^1 P(R) dR. \quad (7.2)$$

Prosječna preciznost je, kao i preciznost, ovisna o omjeru klasa u skupu za testiranje. Odziv i stopa lažnih pozitiva nisu. Krivulja koja opisuje odnos između odziva (stope stvarnih pozitiva) i stope lažnih pozitiva naziva se **ROC-krivulja** (engl. *receiver operating characteristic*). Ona nije ovisna o omjeru klasa. **Površina ispod ROC-krivulje**,

$$AUROC := \int_0^1 FPR(R) dR, \quad (7.3)$$

se može interpretirati kao vjerojatnost da će kod klasifikatora neki pozitivni primjer biti pozitivniji od nekog negativnog primjera. Za nasumični klasifikator je očekivanje te površine $\frac{1}{2}$.

7.1.2. Višeklasna klasifikacija

Kod višeklasne klasifikacije u C klasa, konfuzijska matrica je dimenzija $C \times C$. Njen element s indeksima $[i, j]$ predstavlja broj primjera klasificiranih u klasu i i pripadaju klasi j . Za svaku klasu k možemo definirati

$$\begin{aligned} TP_k &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = k] \mathbb{I}[y_i = k], & FP_k &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) = k] \mathbb{I}[y_i \neq k], \\ FN_k &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) \neq k] \mathbb{I}[y_i = k], & TN_k &:= \sum_i \mathbb{I}[h(\mathbf{x}_i) \neq k] \mathbb{I}[y_i \neq k]. \end{aligned} \quad (7.4)$$

Neke od evaluacijskih mjera za višeklasnu klasifikaciju u C klasa su:

1. **točnost** (engl. *accuracy*) — udio točno klasificiranih primjera: $A := \frac{\sum_k TP_k}{N}$

2. **makro-usrednjena preciznost** (engl. *macro-averaged precision*) — srednja preciznost po klasama: $P_m := \frac{1}{C} \sum_k P_k$, gdje $P_k := \frac{TP_k}{TP_k + FP_k}$
3. **srednja prosječna preciznost** (engl. *mean average precision*) — srednja prosječna preciznost po klasama: $mAP := \frac{1}{C} \sum_k AP_k$, gdje $AP_k := \int_0^1 P_k(R_k) dR_k$
4. **makro-usrednjeni Jaccardov indeks ili srednji omjer presjeka i unije** (engl. *mean intersection over union*) - srednji Jaccardov indeks po klasama: $J_m := mIoU := \frac{1}{C} \sum_k J_k$, gdje $J_k := \frac{TP_k}{TP_k + FP_k + FN_k}$.

7.1.3. Semantička segmentacija

Kod semantičke segmentacije ulazni primjeri su slike i svakom pikselu se dodjeljuje oznaka. Kod evaluacije se obično pikseli iz svih slika iz skupa za ispitivanje smatraju nezavisnim primjerima, pa je zbroj svih elemenata konfuzijske matrice $|\mathcal{D}|HW$, gdje je $|\mathcal{D}|$ broj slika u skupu za ispitivanje, a H i W prostorne dimenzije slika. Kod semantičke segmentacije se često koriste točnost i srednji omjer presjeka i unije.

7.2. Procjena i razlikovanje nesigurnosti kod semantičke segmentacije pomoću MC-dropouta

Aproksimacija bayesovske neuronske mreže pomoću *dropouta* (Gal i Ghahramani, 2015a,a, 2016) je opisana u pododjeljku 6.6.1. Neki pristupi (Kendall i Gal, 2017; Smith i Gal, 2018) za razlikovanje aleatorne i epistemičke nesigurnosti pomoću *dropouta* su opisani u pododjeljku 6.5. Kendall i Gal (2017) su postupak opisan u pododjeljku 6.6.1 koristili na konvolucijskim mrežama i zadatku semantičke segmentacije, gdje se svakom pikselu dodeljuje oznaka, i na zadatku procjene prostorne dubine (regresije) svakog piksela ulazne slike.

Kako bi razlikovali aleatornu i epistemičku nesigurnosti kod semantičke segmentacije, Kendall i Gal (2017) aleatornu nesigurnost modeliraju pomoću predikcije varijanci logita svakog piksela kao što je opisano u pododjeljku 6.5.1, tj. gubitak za svaki piksel je negativni logaritam vjerojatnosti ciljane klase, ali vjerojatnosti se računaju kao očekivanje izlaza softmaxa po razdiobi logita, kao što

je opisano jednađbom (6.17). Očekivanje se procjenjuje *Monte Carlo* aproksimacijom. Za semantičku segmentaciju se ukupan gubitak jedne slike računa kao srednji gubitak po pikselima. Kao mjeru epistemičke nesigurnosti, Kendall i Gal (2017) za svaki piksel koriste entropiju njegove izlazne razdiobe dobivene uz *MC-dropout*.

Ovaj pristup nije do kraja ostvaren i umjesto njega je isproban *MC-dropout* uz procjenu epistemičke nesigurnosti međusobnom informacijom kao što je opisano u pododjeljku 6.5.2 (Smith i Gal, 2018).

Neka \bar{h} označava *Monte Carlo* aproksimaciju hipoteze – usrednjeni izlaz softmaxa:

$$\bar{h}(\mathbf{x}) := \frac{1}{M} \sum_{i=1}^M h(\mathbf{x}; \tilde{\boldsymbol{\theta}}_i), \quad (7.5)$$

gdje je M broj uzoraka za *Monte Carlo* aproksimaciju, a $\tilde{\boldsymbol{\theta}}_i$ uzorci iz varijacijske razdiobe koja odgovara *dropoutu*. Ukupna procjena nesigurnosti predikcije je entropija procjena entropije izlazne razdiobe:

$$H(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) \approx -\bar{h}(\mathbf{x})^\top \ln(\bar{h}(\mathbf{x})). \quad (7.6)$$

Aleatorna nesigurnost se procjenjuje kao uvjetna entropija izlazne razdiobe s obzirom na aposteriornu razdiobu parametara (drugi član u izrazu (6.20)), čemu odgovara aproksimacija prosječnom entropijom uzoraka izlazne razdiobe:

$$\mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} H(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) \approx \frac{1}{M} \sum_{i=1}^M H(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i), \quad (7.7)$$

gdje je

$$H(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i) = -h(\mathbf{x}; \tilde{\boldsymbol{\theta}}_i)^\top \ln(h(\mathbf{x}; \tilde{\boldsymbol{\theta}}_i)) \quad (7.8)$$

entropija razdiobe izlaza softmaxa i -tog uzorka, a epistemička nesigurnost se procjenjuje kao procjena međusobne informacija prema izrazu (6.20), tj. kao razlika ukupne nesigurnosti u izrazu (7.6) i aleatorne nesigurnosti u izrazu (7.7):

$$I(\mathbf{y}, \boldsymbol{\theta} \mid \mathbf{x}, \mathcal{D}) = H(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) - \mathbf{E}_{\boldsymbol{\theta} \mid \mathcal{D}} H(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{x}) \quad (7.9)$$

$$\approx -\bar{h}(\mathbf{x})^\top \ln(\bar{h}(\mathbf{x})) - \frac{1}{M} \sum_{i=1}^M H(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_i). \quad (7.10)$$

Modeli i skupovi podataka

Za ovaj rad su ispitane neke od tih ideja na zadatku semantičke segmentacije. Umjesto mreže koju su koristili Kendall i Gal (2017), korištena je mreža LadderDensenet-121 (Kreso et al., 2017), za koju je potrebno manje memorije. Korištena je vlastita implementacija te mreže koja ne postiže jednako dobru performansu kao originalna iako nije poznata razlika u inicijalizaciji i učenju. Ta mreža će se označavati s *LadderDensenet-121-V*. Na validacijskom skupu Cityscapesa (Cordts et al., 2016) LadderDensenet-121-V bez *dropout*a postiže srednji omjer presjeka i unije $mIoU = 0.6721 \pm 0.007$ i točnost piksela $A = 0.9459 \pm 0.0006$ (sredina i standardna devijacija skupa od 5 mjerenja), što je značajno manje od $mIoU = 0.7282$ i $A = 0.9506$ kod originalne implementacije. Na skupu za testiranje CamVida (Brostow et al., 2008) ista mreža uz isti postupak učenja postiže $mIoU = 0.6774 \pm 0.0031$ i $A = 0.9176 \pm 0.0025$. Pokazalo se da dodavanje *dropout*a s vjerojatnošću isključivanja $1 - p = 0.2$, prema (Huang et al., 2016), loše utječe na performansu modela. Čini se da ni vjerojatnost isključivanja 0.1 nema značajno pozitivan utjecaj na rezultat u odnosu na mrežu bez *dropout*a, pogotovo za Cityscapes, koji je veći skup od CamVida.

Unatoč lošijoj performansi, za eksperimente je, ako nije drugačije navedeno, korištena mreža s vjerojatnošću isključivanja 0.2 kako aposteriorna razdioba parametara ne bi imala premalu entropiju i različiti uzorci mreže s *dropoutom* ne bi bili previše slični, što bi onemogućilo dobro procjenjivanje epistemičke nesigurnosti predikcije, koja proizlazi iz nesigurnosti u parametre. Kako nije bilo dovoljno memorije za učenje mreže s *dropoutom* na slikama iz Cityscapesa umanjenim na dimenzije 1024×448 , kao kod Kreso et al. (2017), pri učenju je korišteno nasumično izrezivanje dijelova slika dimenzija 448×448 i broj epoha je povećan s 30 na 80. Kod CamVida su slike dimenzija 480×360 i bilo je dovoljno memorije. Veličina mini-grupe za CamVid je bila 8.

Za *MC-dropout* je, kao kod Kendall i Gal (2017) za evaluaciju korišteno po 50 uzoraka, tj. unaprijednih prolaza s *dropoutom*, za procjenu izlazne razdiobe klasa za svaki primjer.

Model	$mIoU$	A
DenseNet + <i>dropout</i> (Kendall i Gal, 2017)	0.671	-
+ aleatorna nesigurnost	0.672	-
+ <i>MC-dropout</i>	0.673	-
+ aleatorna nesigurnost i <i>MC-dropout</i>	0.674	-
LadderDenseNet-121-V	0.677	0.918
+ <i>dropout</i> -0.1	0.678	0.913
+ <i>MC-dropout</i> -0.1	0.677	0.915
+ <i>dropout</i> -0.2	0.661	0.912
+ <i>MC-dropout</i> -0.2	0.665	0.913

Tablica 7.1: Usporedba rezultata evaluacije na skupu CamVid. Vrijednosti za LadderDenseNet-121-V bez *dropouta* je prosjek 5 evaluacija, vrijednosti za LadderDenseNet-121-V s običnim *dropoutom* i *MC-dropoutom* su prosjek 2 evaluacije za vjerojatnost isključivanja 0.1 i 2 evaluacije za vjerojatnost isključivanja 0.2.

Rezultati

U tablici 7.1 su prikazani rezultati evaluacije različitih modela učenih na CamVidu (osim podskupa za testiranje). Može se vidjeti da MC-dropout ima pozitivan učinak na evaluaciju u odnosu na obični *dropout*.

Napravljen je eksperiment s učenjem mreže na podskupovima CamVida veličine $\frac{1}{2}$, $\frac{1}{4}$ i $\frac{1}{8}$ cijelog skupa za usporedbu procjena aleatorne i epistemičke nesigurnosti. Smanjivanjem skupa za učenje epistemička nesigurnost trebala bi rasti, a trebao bi rasti i omjer epistemičke i aleatorne nesigurnosti. Rezultati u tablici 7.2 pokazuju da raste procjena epistemičke nesigurnosti, ali omjer procjena epistemičke i aleatorne nesigurnosti ne raste.

CamVid – $1/4$, $1/2$ i cijeli skup za učenje – usporedba aleatorne i epistemičke nesigurnosti kao u Smith i Gal (2018)

kalibracija nesigurnosti – ako stignem

eksplicitna aleatorna nesigurnost s Gaussovom razdiobom logita kao u Kendall i Gal (2017) – ako stignem (mislim da neću)

slike – primjeri izlaza

Učenje	Testiranje	$mIoU$	Aleatorna	Epistemička	$\frac{\text{epistemička}}{\text{aleatorna}}$
CamVid-tv	CamVid-test	0.663	0.207	0.025	0.1206
CamVid-tv/2	CamVid-test	0.615	0.272	0.026	0.0968
CamVid-tv/4	CamVid-test	0.546	0.391	0.029	0.0732
CamVid-tv/8	CamVid-test	0.460	0.538	0.034	0.0633
CamVid-tv	CamVid-val	0.827	0.118	0.011	0.0931
CamVid-tv	WildDash-bench	-	0.501	0.075	0.149

Tablica 7.2: Procjene aleatorne i epistemičke nesigurnosti za različite parove skupa za učenje i skupa za ispitivanje. CamVid-tv označava uniju skupa za učenje i skupa za validaciju, CamVid-tv/ n da se koristi $\frac{1}{n}$ od svih primjera skupa CamVid-tv, CamVid-val skup za validaciju koji je podskup skupa CamVid-tv, CamVid-test skup za testiranje, WildDash skup na kojemu bi se

7.3. Prepoznavanje izvanrazdiobnih i krivo klasificiranih primjera na temelju izlaza softmaxa ili logita kod klasifikacije slika

Isprobani su postupci koji su opisani u odjeljku 6.6.2, tj. prepoznavanje izvanrazdiobnih primjera na temelju maksimalnog izlaza softmaxa (Hendrycks i Gimpel, 2016) i postupak predložen u Liang et al. (2017). Uz to su isprobani analogni postupci kod kojih se klasifikacija provodi na temelju maksimalnog logita – klasifikacija na temelju maksimalnog logita uz pomak i bez pomaka ulaza u smjeru povećanja vjerojatnosti klase maksimalnog izlaza softmaxa prema izrazu (6.30).

Modeli

Za ispitivanje su korištene rezidualna mreža WRN-28-10 (Zagoruyko i Komodakis, 2016) i mreža DenseNetBC (Huang et al., 2016) s dubinom $L = 100$ i faktorom rasta (engl. *growth rate*) $k = 12$, koja će biti označavana s *DN-100-12*. Ni kod jedne mreže se ne koristi *dropout*. Korištene su vlastite implementacije tih mreža koje se ne bi trebale previše razlikovati od originalnih, tj. trebalo bi biti sve isto kao što je opisano u Zagoruyko i Komodakis (2016); Huang et al. (2016). I jedna i druga ipak postižu malo manju točnost od originalnih mreža na skupu za testiranje skupa CIFAR-10 (Krizhevsky, 2009). Od modela korištenih za ispitivanje, WRN-28-10 ima točnost oko 0.957, a DN-100-12 oko 0.948.

Izvanrazdiobni skupovi

Za ove eksperimente je kao unutarrazdiobni skup korišten podskup za testiranje iz skupa CIFAR-10. Slike su dimenzija 32×32 i ima ih 10000 u skupu za testiranje. Za izvanrazdiobne skupove su, slično kao kod [Hendrycks i Gimpel \(2016\)](#); [Liang et al. \(2017\)](#), korišteni skupovi:

1. TinyImageNet¹ — podskup ImageNet-a ([Deng et al., 2009](#)) sa slikama iz 200 klasa. Skup za testiranje sadrži 10000 slika. Za ove eksperimente su iz skupa za testiranje konstruirana dva skupa: TinyImageNet-C, kojeg čine slike iz kojih su nasumično izrezani dijelovi dimenzija 32×32 , i TinyImageNet-R, kojeg čine slike umanjene na iste dimenzije, tj. dimenzije slika iz skupa CIFAR-10.
2. LSUN ([Yu et al., 2015](#)) — skup sličan TinyImageNetu. Skup za testiranje sastoji se od 10000 slika. Kao i za TinyImageNet, iz skupa za testiranje konstruirana su dva skupa: LSUN-C, kojeg čine slike iz kojih su nasumično izrezani dijelovi dimenzija 32×32 , i LSUN-R, kojeg čine slike umanjene slike.
3. iSUN ([Xu et al., 2015](#)) — podskup SUN-a ([Xiao et al., 2016](#)). Za eksperimente se koriste sve slike iSUN-a umanjene na dimenzije 32×32 .
4. Gaussov šum — slučajni uzorci nizova dimenzija $32 \times 32 \times 3$ tako da svaki element ima nezavisnu vrijednost iz Gaussove razdiobe s očekivanjem 0 i varijancom 1.
5. Uniformni šum — slučajni uzorci nizova dimenzija $32 \times 32 \times 3$ tako da svaki element ima nezavisnu vrijednost iz uniformne razdiobe s očekivanjem 0 i varijancom 1.

Kao i kod učenja, za svaki skup, slike koje se daju kao ulaz mreži su normalizirane oduzimanjem prosječne vrijednosti i dijeljenjem sa standardnom devijacijom svake komponente (RGB) prema pikselima u skupu za učenje (i validaciju).

Odabir parametara

Na temelju rezultata u [Liang et al. \(2017\)](#), koji pokazuju da su veće temperature bolje, za eksperimente u kojima se koristi temperaturno skaliranje logita prije softmaxa koristi se temperatura $T = 1000$. Za odabir veličine pomaka ϵ iz svakog izvanrazdiobnog skupa se izvoji 1000 slika. Optimalni ϵ se traži u skupu

¹<https://tiny-imagenet.herokuapp.com/>

	$FPR_{R=0.95}/\%$					$AUROC/\%$					$AP_{in}/\%$					$AP_{out}/\%$					$\epsilon/10^{-3}$				
CIFAR-10	95.0	95.0	95.0	95.0	95.0	50.0	50.0	50.0	50.0	50.0	62.1	52.6	52.6	52.6	52.6	47.4	47.4	47.4	47.4	47.4	0.0	0.0	0.0	0.0	0.0
Gaussian	74.3	6.1	0.0	0.0	0.0	93.3	97.0	98.9	98.9	98.9	96.2	98.3	99.3	99.3	99.3	85.2	91.2	96.8	96.8	96.8	0.0	0.0	2.0	0.0	2.0
LSUN-C	54.0	33.3	33.3	39.5	33.3	91.8	94.5	94.5	93.1	94.5	94.1	95.5	95.5	94.3	95.5	88.0	92.9	92.9	91.3	92.9	0.0	0.0	0.0	0.0	0.0
LSUN-R	48.7	18.9	12.1	15.0	12.1	93.1	96.8	97.7	97.4	97.7	95.1	97.5	98.1	97.9	98.1	89.8	95.9	97.4	96.8	97.4	0.0	0.0	7.0	0.0	7.0
TinyImageNet-C	52.1	23.9	18.8	20.4	18.8	92.5	96.0	96.2	96.3	96.2	94.6	96.8	96.5	96.9	96.5	88.9	94.7	95.8	95.5	95.8	0.0	0.0	6.0	0.0	6.0
TinyImageNet-R	58.5	39.5	36.5	37.0	36.5	90.3	92.4	92.3	92.7	92.3	92.6	93.6	92.9	93.6	92.9	86.2	90.6	91.3	91.4	91.3	0.0	0.0	6.0	0.0	6.0
Uniform	74.3	6.1	0.0	0.0	0.0	93.3	97.0	98.9	98.9	98.9	96.2	98.3	99.3	99.3	99.3	85.2	91.2	96.8	96.8	96.8	0.0	0.0	2.0	0.0	2.0
iSUN	51.1	20.5	12.7	15.5	12.7	92.9	96.6	97.6	97.3	97.6	95.4	97.7	98.2	98.0	98.2	88.3	95.2	97.0	96.3	97.0	0.0	0.0	7.0	0.0	7.0

Tablica 7.3: DN-100-12, CIFAR-10, $A = 0.9481$, $T = 1000$ (softmax, t= logits softmax-perturb logits-perturb).

	$FPR_{R=0.95}/\%$					$AUROC/\%$					$AP_{in}/\%$					$AP_{out}/\%$					$\epsilon/10^{-3}$				
TinyImageNet-C	51.8	35.7	32.0	34.0	32.0	90.6	93.1	93.3	92.3	93.3	92.5	94.1	93.8	92.2	93.8	87.4	91.7	92.5	91.8	92.5	0.0	0.0	4.0	0.0	4.0
TinyImageNet-R	55.0	43.2	42.7	49.8	42.7	89.4	90.2	90.0	85.1	90.0	90.3	90.1	89.6	83.5	89.6	86.0	88.8	88.9	85.2	88.9	0.0	0.0	1.0	0.0	1.0
LSUN-C	37.4	17.2	17.2	37.0	17.2	94.3	96.6	96.6	89.1	96.6	95.6	97.0	97.0	87.6	97.0	91.5	96.0	96.0	90.0	96.0	0.0	0.0	0.0	0.0	0.0
LSUN-R	47.3	28.0	22.3	23.8	22.3	92.6	94.6	94.8	94.1	94.8	94.2	95.2	94.7	93.8	94.7	89.4	93.5	94.6	94.1	94.6	0.0	0.0	4.0	0.0	4.0
iSUN	49.0	31.8	25.8	26.5	25.8	91.8	93.9	94.3	93.4	94.3	94.0	95.0	94.9	93.7	94.9	87.4	91.9	93.3	92.8	93.3	0.0	0.0	3.0	0.0	3.0
Uniform	88.2	71.8	0.0	0.0	0.0	87.4	91.6	98.7	98.7	98.7	92.6	95.0	99.1	99.1	99.1	75.1	82.5	97.8	97.8	97.8	0.0	0.0	7.0	0.0	7.0
Gaussian	88.2	71.8	0.0	0.0	0.0	87.4	91.6	98.7	98.7	98.7	92.6	95.0	99.1	99.1	99.1	75.1	82.5	97.8	97.8	97.8	0.0	0.0	7.0	0.0	7.0

Tablica 7.4: (softmax ($T = 0$), softmax ($T = 1000$), softmax-perturb ($T = 1000$), logits, logits-perturb ($T = 1000$))

WRN-28-10, CIFAR-10, $A = 0.9571$, $T = 1000$, .

$\{0, 10^{-3}, 2 \cdot 10^{-3}, \dots, 12 \cdot 10^{-3}\}$. I kod mreže WRN-28-10 i kod mreže DN-100-12 (vlastitih implementacija), optimalni ϵ je ispadao oko 5 puta veći nego kod [Liang et al. \(2017\)](#) za svaki izvanrazdiobni skup.

Rezultati

Rezultati glavnih eksperimenata su prikazani u tablicama 7.3 i 7.4. Za svaku evaluacijsku mjeru, 5 stupaca redom predstavlja klasifikaciju maksimalne vrijednosti:

1. softmaks uz $T = 1$
2. softmaks uz $T = 1000$
3. softmaks uz $T = 1000$ i pomak ulaza
4. logita (uz bilo koji T)
5. logita uz pomak ulaza za $T = 1000$.

Eksperimenti su pokazali da klasifikacija maksimalnog logita s pomakom ulaza uz $T = 1000$ daje skoro pa iste rezultate kao klasifikacija maksimalne vrijednosti softmaks uz $T = 1000$, tj. nije se vidjela razlika barem u prve 3 decimale. Slijedi pokušaj objašnjenja koji ipak ne objašnjava zašto se softmaks s $T = 1000$ bez pomaka značajnije razlikuje od logita bez pomaka. Neka su $s_i = s_{[i]}$ logiti. Prema

definiciji softmaxa,

$$\text{softmax}\left(\frac{1}{T}\mathbf{s}\right)_{[i]} = \frac{\exp(s_i/T)}{\sum_j \exp(s_j/T)}. \quad (7.11)$$

Za dovoljno velik T eksponencijalne funkcije se mogu linearno aproksimirati s malom greškom, pa

$$\text{softmax}\left(\frac{1}{T}\mathbf{s}\right)_{[i]} \approx \frac{1 + s_i/T}{\sum_j (1 + s_j/T)} = \frac{1 + s_i/T}{C + \frac{1}{T} \sum_j s_j}, \quad (7.12)$$

gdje je C broj klasa. U eksperimentima s mrežama naučenim na skupu CIFAR-10, kod mreže DN-100-12 je zbroj logita skoro uvijek u intervalu $[0.001, 0.004]$, a maksimalni logit u intervalu $[2, 30]$, a kod mreže WRN-28-10 je zbroj logita skoro uvijek u intervalu $[0, 0.0004]$, a maksimalni logit u intervalu $[2, 20]$. Primjeri iz različitih skupova su za DN-100-12 prikazani u prostoru

maksimalni logit – zbroj logita na slici 7.2. Zbrojevi logita podijeljeni s velikom temperaturom T su zanemarivi u odnosu na C , pa softmax možemo aproksimirati ovako:

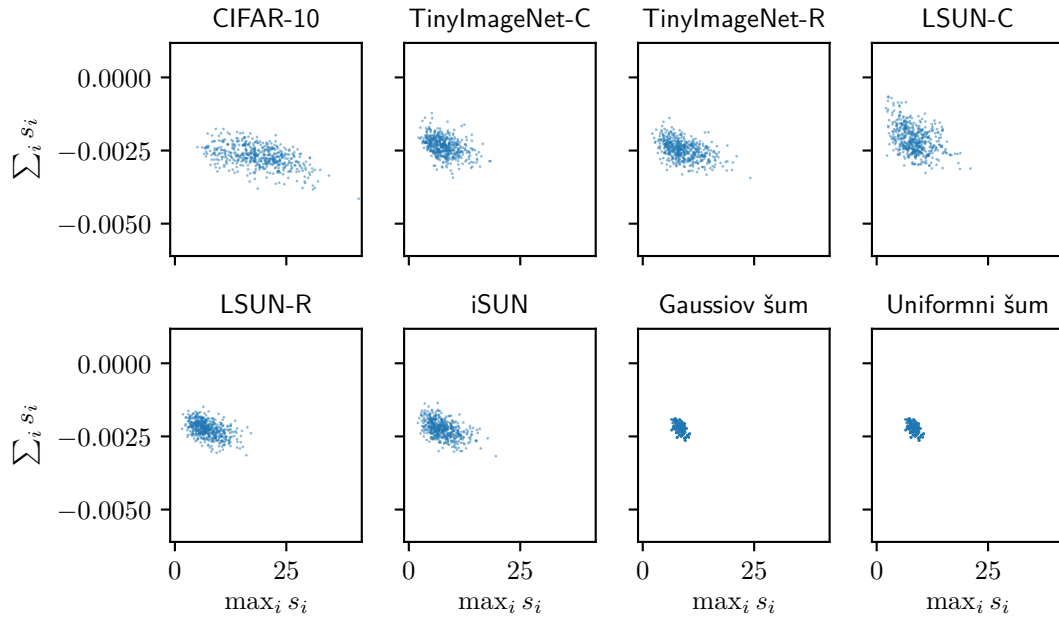
$$\text{softmax}\left(\frac{1}{T}\mathbf{s}\right)_{[i]} \approx \frac{1 + s_i/T}{C} = \frac{1}{C} + \frac{1}{TC} s_i. \quad (7.13)$$

Uz ovakvu aproksimaciju vidimo da se maksimalni izlaz softmaxa svodi na afinu transformaciju maksimalnog logita, pa se klasifikacija maksimalne vrijednosti softmaxa svodi na klasifikaciju maksimalnog logita.

http://elbereth.zemris.fer.hr:8080/wiki_vision2/index.php?title=Uncertainty

	$FPR_{R=0.95}/\%$					$AUROC/\%$					$AP_{in}/\%$					$AP_{out}/\%$				
Gaussiov šum	74.1	8.9	0.0	0.0	0.0	93.1	96.9	99.0	99.0	99.0	96.1	98.3	99.4	99.4	99.4	86.1	91.1	98.3	98.3	98.3
LSUN-C	57.3	35.6	34.8	41.0	34.8	90.8	93.8	92.0	92.4	92.0	93.1	94.8	92.0	93.4	92.0	87.3	92.6	91.9	91.1	91.9
LSUN-R	49.4	19.3	9.2	14.8	9.3	93.2	96.9	98.0	97.6	98.0	95.2	97.6	98.4	98.0	98.4	90.1	96.2	97.7	97.2	97.7
TinyImageNet-C	53.3	26.0	19.3	21.4	19.3	92.5	96.0	96.6	96.5	96.6	94.7	97.0	97.2	97.2	97.2	89.0	95.0	96.2	95.8	96.2
TinyImageNet-R	59.0	40.1	40.4	39.1	40.4	90.4	92.7	92.9	93.0	92.9	92.8	94.1	94.2	94.2	94.2	86.3	91.0	91.5	91.8	91.5
Uniformni šum	74.1	8.9	0.0	0.0	0.0	93.1	96.9	99.0	99.0	99.0	96.1	98.3	99.4	99.4	99.4	86.1	91.1	98.3	98.3	98.3
iSUN	51.4	20.9	12.6	15.2	13.4	92.8	96.4	97.6	97.2	97.6	94.9	97.2	98.0	97.7	98.0	89.5	95.7	97.4	96.8	97.4

	$FPR_{R=0.95}/\%$					$AUROC/\%$					$AP_{in}/\%$					$AP_{out}/\%$					$\epsilon/10^{-3}$				
LSUN-C	66.2	45.3	42.4	50.9	42.4	88.8	92.3	89.1	90.8	89.1	92.2	93.5	89.7	92.3	89.7	83.7	90.4	89.1	88.9	89.1	0.0	0.0	9.0	0.0	9.0
LSUN-R	54.9	26.4	14.2	18.9	14.2	91.6	96.1	97.7	97.0	97.7	94.1	97.0	98.2	97.6	98.2	88.0	95.3	97.4	96.6	97.4	0.0	0.0	9.0	0.0	9.0
TinyImageNet-C	59.3	31.1	29.6	25.8	29.6	90.5	95.1	95.6	96.0	95.6	93.5	96.3	96.6	96.8	96.6	86.0	93.8	94.8	95.4	94.8	0.0	0.0	1.0	0.0	1.0
TinyImageNet-R	64.7	46.2	46.2	42.2	46.2	88.6	91.3	91.3	92.0	91.3	91.8	93.3	93.3	93.6	93.3	83.8	89.3	89.3	90.7	89.3	0.0	0.0	0.0	0.0	0.0
iSUN	59.1	27.1	16.4	19.8	16.4	91.2	95.8	97.4	96.8	97.4	93.9	96.7	97.8	97.4	97.8	87.1	94.8	97.2	96.4	97.2	0.0	0.0	5.0	0.0	5.0
Uniformni šum	80.4	26.7	0.0	0.0	0.0	91.0	95.9	99.0	99.0	99.0	94.8	97.6	99.4	99.4	99.4	82.4	89.2	98.5	98.5	98.5	0.0	0.0	2.0	0.0	2.0
Gaussiov šum	80.4	26.7	0.0	0.0	0.0	91.0	95.9	99.0	99.0	99.0	94.8	97.6	99.4	99.4	99.4	82.4	89.2	98.5	98.5	98.5	0.0	0.0	2.0	0.0	2.0



Slika 7.2: Odnos maksimalnog logita i zbroja logita za različite skupove kod mreže DN-100-12. Svaka točka predstavlja jedan primjer iz podskupova s po 500 primjera iz skupova korištenih za ispitivanje.

T=1 za logite DenseNet	$FPR_{R=0.95}/\%$					$AUROC/\%$					$AP_{in}/\%$					$AP_{out}/\%$					$\epsilon/10^{-3}$				
LSUN-C	55.7	34.3	34.3	40.1	37.0	91.6	94.3	94.3	92.8	90.4	93.9	95.4	95.4	94.1	89.3	87.6	92.6	92.6	91.0	90.4	0.0	0.0	0.0	0.0	9.0
LSUN-R	48.7	18.9	12.1	15.0	13.6	93.1	96.8	97.7	97.4	97.4	95.1	97.5	98.1	97.9	97.7	89.8	95.9	97.4	96.8	96.9	0.0	0.0	7.0	0.0	9.0
TinyImageNet-C	51.8	23.6	18.3	19.9	20.8	92.6	96.1	96.2	96.4	95.4	94.8	96.9	96.5	97.1	95.4	89.0	94.9	95.8	95.6	95.1	0.0	0.0	7.0	0.0	9.0
TinyImageNet-R	58.5	39.5	36.5	37.0	37.2	90.3	92.4	92.3	92.7	91.9	92.6	93.6	92.9	93.6	92.2	86.2	90.6	91.3	91.4	91.0	0.0	0.0	6.0	0.0	6.0
iSUN	51.1	20.5	12.7	15.5	14.1	92.9	96.6	97.6	97.3	97.2	95.4	97.7	98.2	98.0	97.8	88.3	95.2	97.0	96.3	96.5	0.0	0.0	7.0	0.0	9.0
Uniformni šum	74.3	6.1	0.0	0.0	0.0	93.3	97.0	98.9	98.9	99.1	96.2	98.3	99.3	99.3	99.5	85.2	91.2	96.8	96.8	97.6	0.0	0.0	2.0	0.0	3.0
Gaussiov šum	74.3	6.1	0.0	0.0	0.0	93.3	97.0	98.9	98.9	99.1	96.2	98.3	99.3	99.3	99.5	85.2	91.2	96.8	96.8	97.6	0.0	0.0	2.0	0.0	3.0

8. Zaključak

U ovom radu su opisani i eksperimentalno ispitani neki od nadziranih pristupa za procjenu nesigurnosti kod dubokih nadziranih modela.

Jedna skupina razmatranih pristupa se temelji na aproksimaciji bayesovskih neuronskih mreža kod kojih se provodi bayesovska procjena parametara. Oni omogućuju procjenu nesigurnosti i razlikovanje je li uzrok nesigurnosti višeznačnost podatka ili nesigurnost u parametre modela, ali i uz jako jednostavnu varijacijsku razdiobu kojom se aproksimira aposteriora razdioba parametara zahtijevaju puno više računanja u odnosu na uobičajene mreže. Gal i Ghahramani (2016) su učenje mreže s *dropoutom* interpretirali kao varijacijsko zaključivanje. Za ovaj rad je ispitan pristup za procjenjivanje i razlikovanje nesigurnosti pomoću *Monte Carlo dropouta* i drugih ideja iz Kendall i Gal (2017); Smith i Gal (2018).

Druga skupina razmatranih pristupa za procjenu nesigurnosti su pristupi za prepoznavanje primjera koji su izvan razdiobe skupa za učenje Guo et al. (2017); Hendrycks i Gimpel (2016); Liang et al. (2017) na temelju maksimalnog izlaza softmaxa ili logita kod unaprijed naučenih modela. Postupci opisani u Hendrycks i Gimpel (2016); Liang et al. (2017), koji za klasifikaciju koriste maksimalni izlaz softmaxa, uspoređeni su s analognim postupcima koji umjesto maksimalnog izlaza softmaxa koriste maksimalni logit.

LITERATURA

Ethem Alpaydin. *Introduction to Machine Learning*. 2014.

Yoshua Bengio, Olivier Delalleau, i Nicolas Le Roux. The curse of dimensionality for local kernel machines. Technical report, 2005.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 2006.

David M. Blei, Alp Kucukelbir, i Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 2017. URL <http://arxiv.org/abs/1601.00670>.

Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, i Manfred K. Warmuth. Occam's razor. *Inf. Process. Lett.*, 24(6):377–380, Travanj 1987. ISSN 0020-0190. doi: 10.1016/0020-0190(87)90114-1. URL [http://dx.doi.org/10.1016/0020-0190\(87\)90114-1](http://dx.doi.org/10.1016/0020-0190(87)90114-1).

Anselm Blumer, A. Ehrenfeucht, David Haussler, i Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, Listopad 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL <http://doi.acm.org/10.1145/76359.76371>.

Gabriel J. Brostow, Julien Fauqueur, i Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, xx(x):xx–xx, 2008.

Marc C. Kennedy i Anthony O'Hagan. Bayesian calibration of computer models. 2002.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, i Evan Shelhamer. cudnn: Efficient primitives for deep learning. *CoRR*, abs/1410.0759, 2014. URL <http://arxiv.org/abs/1410.0759>.

- Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, i Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. U *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, stranice 303–314, 1989. ISSN 0932-4194. doi: 10.1007/BF02551274. URL <http://dx.doi.org/10.1007/BF02551274>.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, i L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. U *CVPR09*, 2009.
- John S. Denker i Yann LeCun. Transforming neural-net output levels to probability distributions. U *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3, NIPS-3*, stranice 853–859, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. ISBN 1-55860-184-8. URL <http://dl.acm.org/citation.cfm?id=118850.119959>.
- Vincent Dumoulin i Francesco Visin. A guide to convolution arithmetic for deep learning, 2016. URL <http://arxiv.org/abs/1603.07285>.
- Siniša Šegvić. Duboko učenje: Unatražno učenje konvolucijskih slojeva, 2018. URL http://www.zemris.fer.hr/~ssegvic/du/du2convnet_bp.pdf.
- Neven Elezović. *Vjerojatnost i statistika: Slučajne varijable*. 2007.
- Yarin Gal. *Uncertainty in Deep Learning*. Doktorska disertacija, University of Cambridge, 2016.
- Yarin Gal i Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. svezak abs/1506.02142, 2015a. URL <http://arxiv.org/abs/1506.02142>.
- Yarin Gal i Zoubin Ghahramani. Dropout as a Bayesian Approximation: Appendix. 2015b. URL <https://arxiv.org/abs/1506.02157>.
- Yarin Gal i Zoubin Ghahramani. Bayesian convolutional neural networks with Bernoulli approximate variational inference. U *4th International Conference on Learning Representations (ICLR) workshop track*, 2016.

- Xavier Glorot i Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. U Yee Whye Teh i Mike Titterton, urednici, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, svezak 9 od *Proceedings of Machine Learning Research*, stranice 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <http://proceedings.mlr.press/v9/glorot10a.html>.
- Xavier Glorot, Antoine Bordes, i Yoshua Bengio. Deep sparse rectifier neural networks. U Geoffrey J. Gordon, David B. Dunson, i Miroslav Dudík, urednici, *AISTATS*, svezak 15 od *JMLR Proceedings*, stranice 315–323. JMLR.org, 2011. URL <http://dblp.uni-trier.de/db/journals/jmlr/jmlrp15.html#GlorotBB11>.
- Gabriel Goh. Why momentum really works. *Distill*, 2017. doi: 10.23915/distill.00006. URL <http://distill.pub/2017/momentum>.
- Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- Ian J. Goodfellow, Jonathon Shlens, i Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. URL <http://arxiv.org/abs/1412.6572>.
- Peter Grünwald. A tutorial introduction to the minimum description length principle. U *Advances in Minimum Description Length: Theory and Applications*, 2005.
- Ivan Grubišić. Izvještaj: Parsevalove mreže, 2018. URL <https://github.com/Ivan1248/Parseval-networks/blob/master/report/izvjestaj.pdf>.
- Chuan Guo, Geoff Pleiss, Yu Sun, i Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017. URL <http://arxiv.org/abs/1706.04599>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL <http://arxiv.org/abs/1603.05027>.

- Dan Hendrycks i Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016. URL <http://arxiv.org/abs/1610.02136>.
- Geoffrey Hinton. Neural networks for machine learning, lecture 6a: Overview of mini-batch gradient descent. 2012. URL http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.
- Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, i Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL <http://arxiv.org/abs/1207.0580>.
- Gao Huang, Zhuang Liu, i Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL <http://arxiv.org/abs/1608.06993>.
- Sergey Ioffe i Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL <http://arxiv.org/abs/1502.03167>.
- Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, i Lawrence K. Saul. An introduction to variational methods for graphical models. 1999.
- Alex Kendall i Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977, 2017. URL <http://arxiv.org/abs/1703.04977>.
- Diederik P. Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- Armen Der Kiureghian i Ove Ditlevsen. Aleatory or epistemic? Does it matter? 2009.
- Ivan Kreso, Sinisa Segvic, i Josip Krapac. Ladder-style densenets for semantic segmentation of large natural images. U *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- Alexey Kurakin, Ian J. Goodfellow, i Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL <http://arxiv.org/abs/1611.01236>.

- Yann LeCun, Yoshua Bengio, i Geoffrey E. Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015. doi: 10.1038/nature14539. URL <https://doi.org/10.1038/nature14539>.
- Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, i Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, stranice 861–867, 1993. URL <http://dblp.uni-trier.de/db/journals/nn/nn6.html#LeshnoLPS93>.
- Xiang Li, Shuo Chen, Xiaolin Hu, i Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. *CoRR*, abs/1801.05134, 2018. URL <http://arxiv.org/abs/1801.05134>.
- Shiyu Liang, Yixuan Li, i R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017. URL <http://arxiv.org/abs/1706.02690>.
- Yanpei Liu, Xinyun Chen, Chang Liu, i Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. URL <http://arxiv.org/abs/1611.02770>.
- David J. C. MacKay. Bayesian methods for adaptive models, 1992a.
- David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, Svibanj 1992b. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL <http://dx.doi.org/10.1162/neco.1992.4.3.448>.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, i Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017. URL <http://arxiv.org/abs/1706.06083>.
- Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, i Shin Ishii. Distributional smoothing by virtual adversarial examples. *CoRR*, abs/1507.00677, 2015. URL <http://arxiv.org/abs/1507.00677>.
- Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, i Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *CoRR*, abs/1704.03976, 2017. URL <http://arxiv.org/abs/1704.03976>.

- Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, i Pascal Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016. URL <http://arxiv.org/abs/1610.08401>.
- Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. 2012.
- Iain Murray i Zoubin Ghahramani. A note on the evidence and Bayesian Occam's razor. 2005.
- Jan Šnajder. Strojno učenje: 7. logistička regresija ii, 2017. URL http://www.fer.unizg.hr/_download/repository/SU-2017-07-LogistickaRegresija2.pdf.
- Jan Šnajder i Bojana Dalbelo Bašić. *Strojno učenje*. 2014.
- Radford M. Neal. Bayesian learning for neural networks, 1995.
- Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. 2014.
- Anh Mai Nguyen, Jason Yosinski, i Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. U *CVPR*, stranice 427–436. IEEE Computer Society, 2015. ISBN 978-1-4673-6964-0. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#NguyenYC15>.
- Christopher Olah. Calculus on computational graphs: Backpropagation, 2015a. URL <http://colah.github.io/posts/2015-08-Backprop/>.
- Christopher Olah. Visual information theory, 2015b. URL <http://colah.github.io/posts/2015-09-Visual-Information/>.
- Samuel Rathmanner i Marcus Hutter. A philosophical treatise of universal induction. *CoRR*, abs/1105.5721, 2011. URL <http://arxiv.org/abs/1105.5721>.
- Amrith Rawat, Martin Wistuba, i Maria-Irina Nicolae. Adversarial phenomenon in the eyes of bayesian deep learning. *arXiv preprint arXiv:1711.08244*, 2017. URL <https://arxiv.org/abs/1711.08244>.
- Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. URL <http://arxiv.org/abs/1609.04747>. cite arxiv:1609.04747Comment: 12 pages, 6 figures.

- D. E. Rumelhart, G. E. Hinton, i R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. poglavlje Learning Internal Representations by Error Propagation, stranice 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. URL <http://dl.acm.org/citation.cfm?id=104279.104293>.
- Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL <https://ieeexplore.ieee.org/document/6773024/>.
- Lewis Smith i Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *CoRR*, abs/1803.08533, 2018. URL <http://arxiv.org/abs/1803.08533>.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, i Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- Ilya Sutskever. Training recurrent neural networks. *University of Toronto, Toronto, Ont., Canada*, 2013. URL http://www.cs.utoronto.ca/~ilya/pubs/ilya_sutskever_phd_thesis.pdf.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, i Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL <http://arxiv.org/abs/1312.6199>.
- Twan van Laarhoven. L2 regularization versus batch and weight normalization. *CoRR*, abs/1706.05350, 2017. URL <http://arxiv.org/abs/1706.05350>.
- D. Randall Wilson i Tony R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Netw.*, 16(10):1429–1451, Prosinac 2003. ISSN 0893-6080. doi: 10.1016/S0893-6080(03)00138-2. URL [http://dx.doi.org/10.1016/S0893-6080\(03\)00138-2](http://dx.doi.org/10.1016/S0893-6080(03)00138-2).
- Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, i Aude Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1):3–22, Kolovoz 2016. ISSN 0920-5691. doi: 10.1007/s11263-014-0748-y. URL <http://dx.doi.org/10.1007/s11263-014-0748-y>.

Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, i Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *CoRR*, abs/1504.06755, 2015. URL <http://arxiv.org/abs/1504.06755>.

Xitong Yang. Understanding the Variational Lower Bound, 2017. URL <http://legacydirs.umiacs.umd.edu/~xyang35/files/understanding-variational-lower.pdf>.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, i Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. URL <http://dblp.uni-trier.de/db/journals/corr/corr1506.html#YuZSSX15>.

Sergey Zagoruyko i Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL <http://arxiv.org/abs/1605.07146>.

Nadzirani pristupi za procjenu nesigurnosti predikcija dubokih modela

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.