Zkouška

Questions@:, Lecture 1 Questions

- Explain why we need separate train and test data? What is generalization and how the concept relates to underfitting and overfitting? **[10]**
  - Model se může naučit "nazpamět" všechna možná data a proto je potřeba testovat model na jiných než trénovacích datec. Pokud bychom ho neověřovali na separátních datech, mohlo bychom dojít k mylné myšlence, že je náš model super perfektní a na dalších datech, z vnějšího světa, by selhal.
  - Generalizace je zobecnění modelu na trénovacích datech, tak moc abychom na dalších vstupních datech mohli vidět co nejvíce společných znaků. *Generalization refers to your model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.*
  - Underfitting - model není dostatečně naučen, overfitting - model je až moc dobrý na trénovacích datech, přeučený, selhává na všech ostatních.
  - Příliš mnoho generalizace vede k UF, příliš málo generalizace vede k OF
  - níže dodal Kuba:
  - Naším cílem je dosáhnout co nejvyšší výkonnosti pro daný problém, spíše chceme řešit problém pro neznámá data.
  - Snažíme se tedy o nejlepší generalizaci, tedy výkon našeho modelu na neznámých datech.
  - Pokud budeme chtít ověřit funkčnost našeho modelu, nemůžeme měřit výkon našeho modelu na trénovacích datech, na datech na která se model optimalizoval. Je zapotřebí použít separátní testovací data, která jsou ze stejné distribuce, nezávislá na trénovacích.
  - Model tedy budeme učit na trénovacích datech, měřit výkon na testovacích. V návaznosti s generalizací mohou nastat 2 situace, kterých se chceme vyvarovat.
  - Underfitting - model má ještě prostor na to, aby se naučil více obecných rysů, trénovací chyba může dále klesat i testovací

- Overfitting - model se přeučil, až moc se optimalizoval na trénovacích datech a naučil se tedy i rysy, které souvisejí jenom s datama, která viděl. Tímto způsobem selže na testovacích datech. Trénovací chyba stále klesá, testovací se již zvyšuje.
- V obou případech máme slabší generalizaci, než kdybychom byli někde uprostřed. Tedy jsme nad underfittingem a pod overfittingem.

- Define prediction function of a linear regression model and write down L^2-regularized mean squared error loss. **[10]**
  - **predikce** funkce:
  - $$y(\boldsymbol{x}, \boldsymbol{w}, b) =< \boldsymbol{x}, \boldsymbol{w} > +b = \boldsymbol{x}^T\boldsymbol{w} + b$$
  - **normální MSE**:
  - $$\mathrm{MSE}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}(y(\boldsymbol{x}_i, \boldsymbol{w}) - t_i)^2$$
  - suma přes data mocninu (předpověď pro dato - target)
  - L_2 **regularizace**, kde λ je parametr regularizace:
  - $$\mathrm{MSE}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{N}(y(\boldsymbol{x}_i, \boldsymbol{w}) - t_i)^2 + \frac{\lambda}{2}||w||^2$$
  - je to suma čtverců, průměr je $1/n$, když se minimalizuje, tak na konstantě nezáleží, proto se používá sum čtverců, ta dvojka pak hezky vypadne

- Starting from unregularized sum of squares error of a linear regression model, show how the explicit solution can be obtained, assuming $\boldsymbol{X}^T\boldsymbol{X}$ is regular. **[20]**
  - frajeři úlohy za dvacet nedělaj
  - bezchybný důkaz (d)opsal Vojta:
  - Pro nalezení minima $\frac{1}{2}\sum_i^N(\boldsymbol{x}_i^T\boldsymbol{w} - t_i)^2$ se podíváme na hodnoty, kde je derivace error funkce rovna nule vzhledem ke všem vahám w_j.
  - $$\frac{\partial}{\partial w_j}\frac{1}{2}\sum_i^N(\boldsymbol{x}_i^T\boldsymbol{w} - t_i)^2 = \frac{1}{2}\sum_i^N\left(2(\boldsymbol{x}_i^T\boldsymbol{w} - t_i)x_{ij}\right) = \sum_i^N x_{ij}(\boldsymbol{x}_i^T\boldsymbol{w} - t_i)$$
  - Tedy pro všechna j chceme: $\sum_i^N x_{ij}(\boldsymbol{x}_i^T\boldsymbol{w} - t_i) = 0$.
  - Přepíšeme na: $\boldsymbol{X}_{*,j}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) = 0$, nyní přepíšeme rovnosti tak, aby platila pro všechna j najendou pomocí maticové notace $\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) = \boldsymbol{0}$.
  - To se dá přepsat jako:
  - $$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{t}.$$
  - Za zadnání víme, že $\boldsymbol{X}^T\boldsymbol{X}$ je regulární matice, můžeme spočíst její invers, ve výsledku máme:
  - $$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$$
  - $\square$

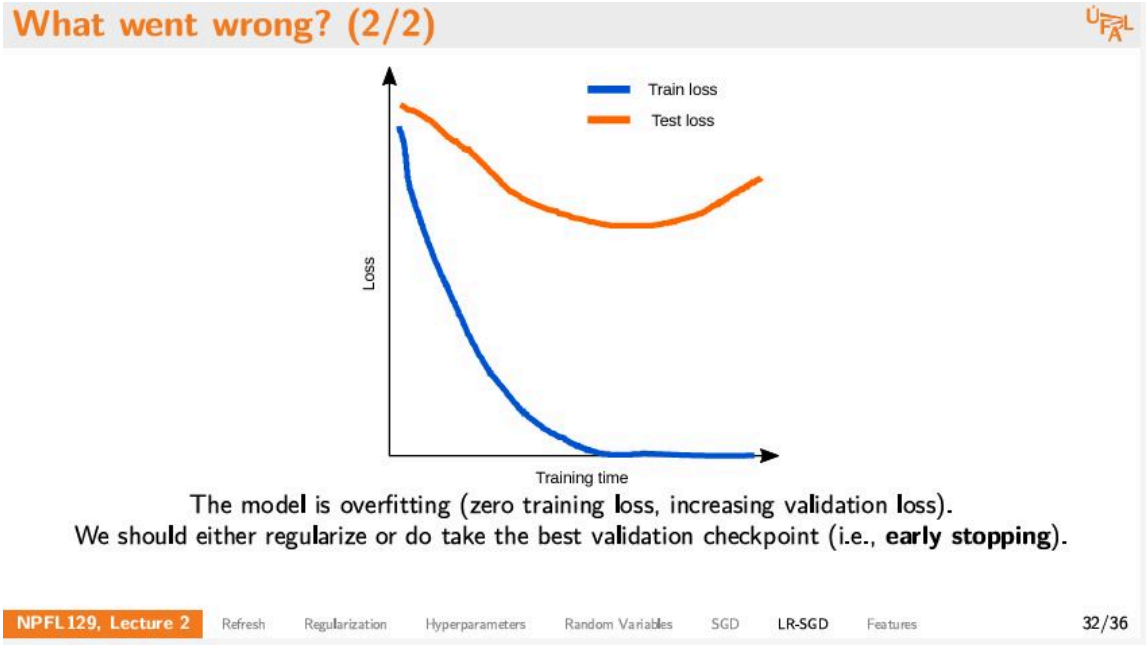
Questions@:, Lecture 2 Questions

- Describe standard gradient descent and compare it to stochastic (i.e., online) gradient descent and minibatch stochastic gradient descent. **[10]**
    - **GD update function**: $w \leftarrow w - \alpha \nabla_w E(w)$, E je error function, α je learning rate, a ta nabla je derivace vah přes všechny proměnné (gradient)
    - minimalizujeme error funkci, iterativně upravujeme váhu. Pomocí gradientu zjistíme směr, ve kterém se zmenší chyba.
    - potřebujeme tedy spočítat gradient, standard gradient používá všechna data
    - **Standard**: použijí se všechna data na spočtení $\nabla_w E(w)$
    - **Stochastic (Online)**: použijeme jeden náhodný prvek z trénovacích dat, odhad je unbiased ale hodně noisy
    - stochastic pouze jeden náhodné dato, jednotlivě obsahuje velký šum, ale limitně dojdeme k podobné střední hodnotě jako standardní (do optima)
    - **Minibatch SGD**: něco mezi Standard a Stochastic, pro spočtení $\nabla_w E(w)$ se použije náhodná nezávislá batch (prvky) o velikosti B
    - **compare**: úspora paměti, rychlejší, stejně nechceme úplné optimum, chceme generalizaci
- Write an L^2-regularized minibatch SGD algorithm for training a linear regression model, including the explicit formulas of the loss function and its gradient. **[10]**
    - lecture 2, slajd 28/36
    - **Input**: dataset ($\boldsymbol{X} \in \mathbb{R}^{N \times D}, \boldsymbol{t} \in \mathbb{R}^N$), learning rate $\alpha \in \mathbb{R}^+$, L^2 síle regularizace $\lambda \in \mathbb{R}$
    - **Output**: váhy $\boldsymbol{w} \in \mathbb{R}^D$
    - **Algoritmus**:
    - $\boldsymbol{w} \leftarrow \mathbf{0}$ (nebo náhodná inicializace)
    - opakuj, dokud nekonverguje (nebo nedojde trpělivost)
        - nasampluj minibatch dat s indexy $\mathcal{B}$ (existuje několik přístupů k samplování, více je na slajdu 28 z lekce 2)
        - $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \frac{1}{|\mathbb{B}|} \sum_{i \in \mathbb{B}} \left(\left(\boldsymbol{x}_i^T \boldsymbol{w} - t_i \right) \boldsymbol{x}_i \right) - \alpha \lambda \boldsymbol{w}$
- Does the SGD algorithm for linear regression always find the best solution on the training data? If yes, explain under what conditions it happens, if not explain why it is not guaranteed to converge. **[20]**
    - Kuba dodal níže, ale moc neví:
    - úplně nejlepší najde pouze v případě, že funkce je konvexní na celém svém definičním oboru (nebo v případě, že stastně začneme tak, že dokonvergujeme do globálního minima) a learning rate splňuje následující podmínky:
        - pořád skáčeme, tedy nezastavíme se, suma přes všechna α je nekonečno
        - jednotliví skoky se rozumně zkracují, suma přes všechna α na druhou je menší než nekonečno, neboli α se zmenšují
        - α jsou nezáporné, nesmíme se vracet
        - (Vojta: těmhle podmínkám teda moc nerozumim)
    - tím, že funkce je konvexní, dostaneme-li se k lokálnímu minimu, dostaneme se k globálnímu
    - ovšem musíme se tam nějak dostat, proto α musí splňovat ty podmínky, tedy, že jsme ochotni skákat dostatečně dlouho a skoky se zkracují

- funkce může mít několik lokálních minim, k jednomu z nich se jistě dostaneme (když algoritmus poběží dost dlouho), ovšem lokální maximum nemusí být globální

- After training a model with SGD, you ended up with a low training error and a high test error. Using the learning curves, explain what might have happen and what steps you might take to prevent this from happening. **[10]**

  - pro přesné určení problému je potřeba celý obrázek, ale jestli se chová model rozumně (časem klesá, zatímco testovací roste), tak jde o OF
  - v případě SGD použijeme tzv. early stopping, jakmile zaznamenáme trend OF, skončíme
  - případně můžeme trénování lépe regularizovat



- You were provided with a fixed training set and a fixed test set and you are supposed to report model performance on that test set. You need to decide what hyperparameters to use. How will you proceed and why? **[5]**

  - na testovací data budeme sahat jen v případě, že testujeme výkonnost modelu
  - na trénování je potřeba znát hyperparametry předem, například pro SGD to jsou batch size, learning rate, pro L_2 regulatizaci nastavíme sílu regularizace
  - nejdřív zvolím parametry, natrénuji model na trénovacích datech, model otestuji na testovacích datech
  - na hledání vhodných parametrů z nějaké množiny (chytře zvolené) lze použít **Gridsearch**
  - z trénovacích dat před použitím Gridsearche vyjmeme data validační, na těch se testují kombinace parametrů
  - myslím, že tohle gridsearch dělá automaticky, přes cv parametr (jo, dělá, ale snad na cvikách to padlo skoro přesně takhle, Vojta)

- What method can be used for normalizing feature values? Explain why it is useful. **[5]**

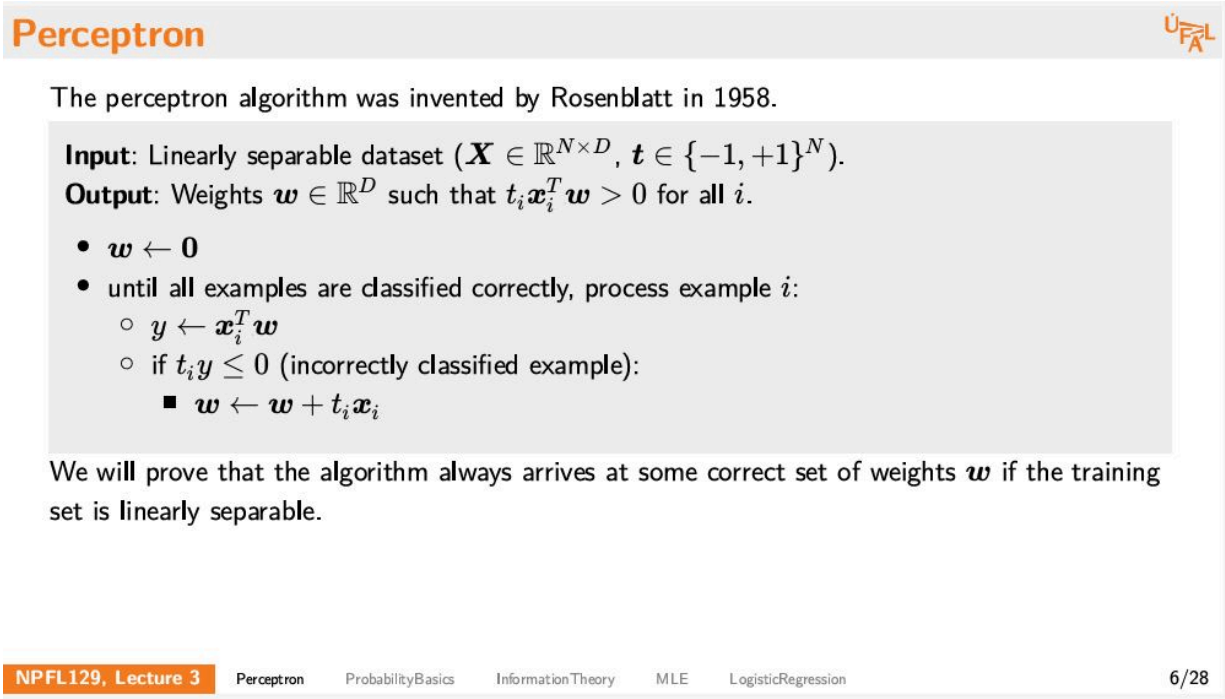
  - **normalization**:
  - $$x_{i,j}^{\text{norm}} = \frac{x_{i,j} - \min_k x_{k,j}}{\max_k x_{k,j} - \min_k x_{k,j}}$$
  - **standardization**:

- $$x_{i,j}^{standard} = \frac{x_{i,j} - \text{mean}(x_j)}{\text{standardní odchylka (standard deviation aka)}\sigma_j}$$
  - ficury v různých škálách by potřebovaly různé learning raty (takhle můžem doufat, že bude stačit všem stejny learning rate)
  - představte si, že jedna featura má hodnoty v rozmezí od 0.00001 az 0.00002 jiná featura v milionech - stejný learning rate udělá velký rozdíl v jednom případě, v druhém případě skoro žádnou

Questions@:, Lecture 3 Questions

- Define binary classification, write down the perceptron algorithm and show how a prediction is made for a given example. **[10]**
  - **binární klasifikace** je funkce, která pro daný vstup předpoví jednu, nebo druhou možnou třídu
  - **prediction**: pro classy $\{-1, 1\}$ je predikce $= \text{sign}(x^T w)$, x je dato, w jsou spočtené váhy z modelu
  - cílem perceptronu je najít takové váhy w, že:

$$\text{sign}(y(\boldsymbol{x}_i; \boldsymbol{w})) = \text{sign}(\boldsymbol{x}_i^T \boldsymbol{w}) = t_i$$



- For discrete random variables, define entropy, cross-entropy, Kullback-Leibler divergence, and prove the Gibbs inequality (i.e., that KL divergence is non-negative). **[20]**
  - **entropy**:
  - $$H(P) = -E_{xp}[log(P(X))]$$
  - **cross-entropy**:
  - $$H(P, Q) = -E_{xp}[log(Q(x))]$$
  - **Kullback-Leibler divergence**:

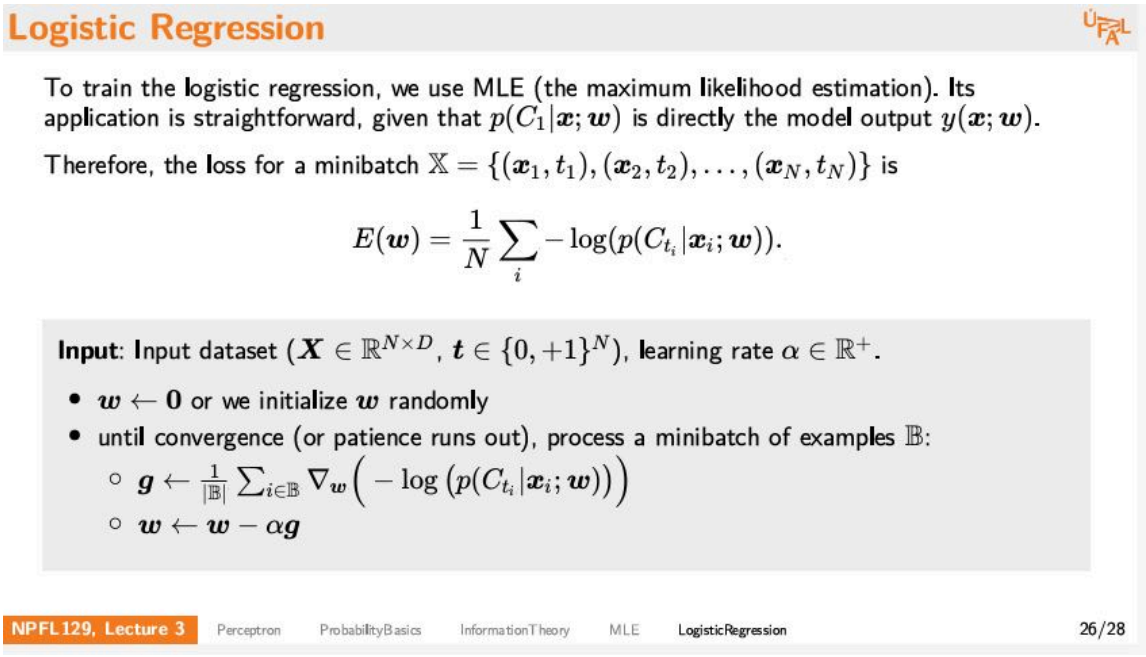
    - $$D_{KL}(P||Q) = H(P,Q) - H(P)$$
    - chceme
    - $$H(P,Q) \geq H(P)$$
    - neboli nezápornost divergence
    - $$H(P,Q) - H(P) \geq 0$$
    - důkaz slajd 15

- Explain the notion of likelihood in maximum likelihood estimation. **[5]**
    - je to pravděpodobnost dat, pravděpodobnost toho, že náš model vygeneroval trénovací data
    - my chceme ale vygenerovat stejné targety
    - můžeme použít MLE s podmínkou
    - chceme target za podmínky toho, že máme data, váhy
    - Lenka: jsem z pana Straky trochu zmatená, říká:
    - pokud jsou váhy pevné, náš model vrací distribuci $p_{model}(\boldsymbol{x}; \boldsymbol{w})$ (např. panna 30 %, orel 70 %)
    - můžeme se na to ale dívat i tak, že data jsou zafixovaná - když dosadíme váhy, můžeme si spočítat, jak moc jsou data pravděpodobná s těmito vahami
    - likelihood je
    - $$L(\boldsymbol{w}) = p_{model}(\boldsymbol{X}; \boldsymbol{w}) = \prod_{i=1}^{N} p_{model}(\boldsymbol{x}_i; \boldsymbol{w})$$
    - likelihood není distribuce, protože se nesčítá do jedničky
    - maximum likelihood estimation vah w - hledám takové váhy, aby pravděpodobnost mých dat s těmito vahami byla co největší
    - Kuba ještě jednou:
    - máme fixní data {x1, x2, ..., xn} = X, váhy w je nějaký parametry
    - takže pro dané wáhy w, nám likelihood L(w) říká, pravděpodobnost dat X, když máme takový model(váhy)
    - to znamená jaká je pravděpodobnost, že mám x1 a x2 a x3 a ... xn, tím, že data jsou z normální distribuce, jsou nezávislé a proto pravděpodobnost, že dostaneme x1 a x2 a x3 a ... xn, je součin pravděpodobností.

- Describe maximum likelihood estimation, as minimizing NLL, cross-entropy, and KL divergence. **[20]**
    - TODO: prezentace 21/29

- Considering binary logistic regression model, write down its parameters (including their size) and explain how prediction is performed (including the formula for the sigmoid function). Describe how we can interpret the outputs of the linear part of the model as logits. **[10]**
    - **sigmoid** function:
    - $$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- **predikce**: $y(x; w) = \sigma(x^T w)$
- **parametry**: batch size B, learning rate α (kladné reálné číslo), počet iterací, vstupní dataset



- **logit**: logaritmus šancí pravděpobností dvou tříd $= \log(\frac{p(C_1|x)}{p(C_0|x)})$
- logaritmus poměrů pravděpodobností, když logit vyjde 1, pak nastane 10krát pravděpodobněji; záporné číslo $\rightarrow$ že nevyjde, je pravděpodobnější
- je to $(x^T w)$, schovaný bias
- lineární část je $\bar{y}(x; w) = x^T w$
- tj. je to nějaké reálné číslo, které se hodí do sigmoidu, ten z něj pak spočte pravděpodobnost
- tj. je to nějaké reálné číslo, které se hodí do sigmoidu, ten z něj pak spočte pravděpodobnost
- otázkou je, jestli tohle číslo má nějakou interpretaci

$$p(C_1|\boldsymbol{x}) = \sigma(y(\boldsymbol{x}; \boldsymbol{w})) = \frac{1}{1 + e^{-\bar{y}(\boldsymbol{x}; \boldsymbol{w})}}$$

- pojďme se juknout, co nám vypadne, když si vyjádříme $\bar{y}$: $p(C_1|\boldsymbol{x}) = \frac{1}{1+e^{-\bar{y}}}$

$$(1 + e^{-\bar{y}})p(C_1|\boldsymbol{x}) = 1$$

$$e^{-\bar{y}} p(C_1|\boldsymbol{x}) = 1 - p(C_1|\boldsymbol{x})$$

$$e^{-\bar{y}} = \frac{p(C_0|\boldsymbol{x})}{p(C_1|\boldsymbol{x})}$$

$$\bar{y} = \log \frac{p(C_1|\boldsymbol{x})}{p(C_0|\boldsymbol{x})}$$

- ano, má svou interpretaci, říká se mu logit a je to logaritmus poměru pravděpodobností
- např. když $\bar{y} = 1$, pak je C_1 e-krát pravděpodobnější než C_0

- Write down an L^2-regularized minibatch SGD algorithm for training a binary logistic regression model, including the explicit formulas of the loss function and its gradient. **[20]**
    - Kuba:
    - slajd 4/34 lec 4
    - 5/34 lec 4
    - zjistit gradient g
    - použít sgd, takže updatovat váhu w <- w - alpha*g + l2 regul
    - zjisti explicitni rovnici pro loss function and gradient

Questions@:, Lecture 4 Questions

- Define mean squared error and show how it can be derived using MLE. **[10]**
    - **normální MSE**:
    - $$\mathrm{MSE}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}(y(\boldsymbol{x}_i, \boldsymbol{w}) - t_i)^2$$
    - Kuba:
    - ehm když koukám na další slajdy tak je to asi ln(e) = 1, ne log(e)

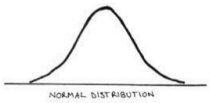
Mean Square Error as MLE

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

Therefore, assume our model generates a distribution $p(t|\boldsymbol{x};\boldsymbol{w}) = \mathcal{N}(t; y(\boldsymbol{x};\boldsymbol{w}), \sigma^2)$.

Now we can apply the maximum likelihood estimation and get

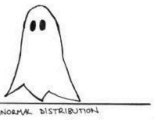
log nezmění hledání maxima



$$\begin{aligned}
\arg\max_{\boldsymbol{w}} p(\boldsymbol{t}|\boldsymbol{X};\boldsymbol{w}) &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{N} -\log p(t_i|\boldsymbol{x}_i;\boldsymbol{w}) \\
&= \arg\min_{\boldsymbol{w}} -\sum_{i=1}^{N} \log \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(t_i - y(\boldsymbol{x}_i;\boldsymbol{w}))^2}{2\sigma^2}} \\
&= \arg\min_{\boldsymbol{w}} -N\log(2\pi\sigma^2)^{-1/2} - \sum_{i=1}^{N} -\frac{\left(t_i - y(\boldsymbol{x}_i;\boldsymbol{w})\right)^2}{2\sigma^2} \\
&= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{N} \frac{\left(t_i - y(\boldsymbol{x}_i;\boldsymbol{w})\right)^2}{2\sigma^2} = \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{i=1}^{N} \left(y(\boldsymbol{x}_i;\boldsymbol{w}) - t_i\right)^2.
\end{aligned}$$

$\log a \cdot b = \log a + \log b$

$\log e^{10} = 10 \cdot \log e$



vymazaly konstanty

vzdhe log e

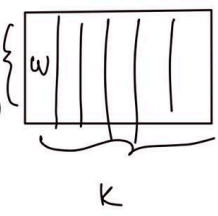
- Considering K-class logistic regression model, write down its parameters (including their size) and explain how prediction is performed (including the formula for the softmax function). Describe how we can interpret the outputs of the linear part of the model as logits. **[10]**

To extend the binary logistic regression to a multiclass case with K classes, we:

- generate K outputs, each with its own set of weights, so that for $\boldsymbol{W} \in \mathbb{R}^{D\times K}$

$$\bar{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{W}) = \boldsymbol{x}^T\boldsymbol{W}, \quad \text{or in other words,} \quad \bar{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{W})_i = \boldsymbol{x}^T(\boldsymbol{W}_{*,i})$$

- generalize the sigmoid function to a softmax function, such that



    - $$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$
    - **softmax** function:

- $$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$p(C_i|x;W) = \frac{e^{(\bar{y}(x;W))_i}}{\sum_j e^{(\bar{y}(x;W))_j}}$ $\quad \log\frac{a}{b} = \log a - \log b$

$\log(p(C_i|x;W)) + c = \bar{y}(x;W) \cdot \ln e$ $\quad 1$

Multiclass Logistic Regression

Using the softmax function, we naturally define that

$$p(C_i|\boldsymbol{x};\boldsymbol{W}) = \boldsymbol{y}(\boldsymbol{x};\boldsymbol{W})_i = \text{softmax}\left(\bar{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{W})\right)_i = \text{softmax}(\boldsymbol{x}^T\boldsymbol{W})_i = \frac{e^{(\boldsymbol{x}^T\boldsymbol{W})_i}}{\sum_j e^{(\boldsymbol{x}^T\boldsymbol{W})_j}}.$$

Considering the definition of the softmax function, it is natural to obtain the interpretation of the linear part of the model $\bar{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{W})$ as **logits** by computing a logarithm of the above:

$$\bar{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{W})_i = \log(p(C_i|\boldsymbol{x};\boldsymbol{W})) + c.$$

- Z toho vyplývá, že logit je log pravděpodobnost dané třídy
- Je tam ale navíc nějaká konstanta c, to je daný tím, že vlastně na spočítání pravděpodobnosti poslední třídy nám stačí n - 1.
- **predikce**:

$$y(\boldsymbol{x};\boldsymbol{W}) = softmax(\boldsymbol{x}^T\boldsymbol{W})$$

- Explain the relationship between the sigmoid function and softmax. **[5]**
  - softmax je zobecnění sigmoidu:
  - $$\sigma(x) = \text{softmax}([x\ 0])_0 = \frac{e^x}{e^x + e^0} = \frac{e^x/e^x}{(e^x + e^0)/e^x} = \frac{1}{1 + e^{-x}}$$
- Write down an L^2-regularized minibatch SGD algorithm for training a K-class logistic regression model, including the explicit formulas of the loss function and its gradient. **[20]**

Multiclass Logistic Regression

To train K-class classification, analogously to the binary logistic regression we can use MLE and train the model using minibatch stochastic gradient descent:

Input: Input dataset ($\boldsymbol{X} \in \mathbb{R}^{N\times D}$, $\boldsymbol{t} \in \{0, 1, \ldots, K-1\}^N$), learning rate $\alpha \in \mathbb{R}^+$.
Model: Let $\boldsymbol{w}$ denote all parameters of the model (in our case, the parameters are a weight matrix $\boldsymbol{W}$ and maybe a bias vector $\boldsymbol{b}$).

- $\boldsymbol{w} \leftarrow \boldsymbol{0}$ or we initialize $\boldsymbol{w}$ randomly
- until convergence (or patience runs out), process a minibatch of examples $\mathbb{B}$:
  - $\boldsymbol{g} \leftarrow \frac{1}{|\mathbb{B}|}\sum_{i\in\mathbb{B}} \nabla_{\boldsymbol{w}}\Big(-\log\left(p(C_{t_i}|\boldsymbol{x}_i;\boldsymbol{w})\right)\Big)$
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha\boldsymbol{g}$

- Prove that decision regions of a multiclass logistic regression **are** convex. **[10]**
    - absolutně netuším, co tím autor zamýšlel

Multiclass Logistic Regression

Note that the decision regions of the binary/multiclass logistic regression are convex (and therefore connected).

To see this, consider $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ in the same decision region R_k.

Any point $\boldsymbol{x}$ lying on the line connecting them is their convex combination, $\boldsymbol{x} = \lambda\boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B$, and from the linearity of $\bar{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{W}$ it follows that

$$\bar{\boldsymbol{y}}(\boldsymbol{x}) = \lambda\bar{\boldsymbol{y}}(\boldsymbol{x}_A) + (1-\lambda)\bar{\boldsymbol{y}}(\boldsymbol{x}_B).$$

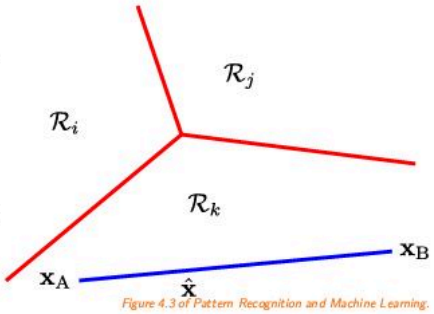


Figure 4.3 of Pattern Recognition and Machine Learning.

Given that $\bar{\boldsymbol{y}}(\boldsymbol{x}_A)_k$ was the largest among $\bar{\boldsymbol{y}}(\boldsymbol{x}_A)$ and also given that $\bar{\boldsymbol{y}}(\boldsymbol{x}_B)_k$ was the largest among $\bar{\boldsymbol{y}}(\boldsymbol{x}_B)$, it must be the case that $\bar{\boldsymbol{y}}(\boldsymbol{x})_k$ is the largest among all $\bar{\boldsymbol{y}}(\boldsymbol{x})$.

- Considering a single-layer MLP with D input neurons, H hidden neurons, K output neurons, hidden activation f, and output activation a, list its parameters (including their size) and write down how the output is computed. **[10]**
    - parametry, W a b? Napiš jejich velikost něco jako W(h) by měla velikost DxH, W(y) -> HxD, b(h) -> 1xH

Multilayer Perceptron

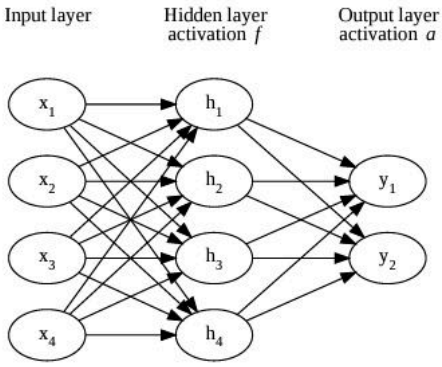
We now extend the model by adding a **hidden layer** with activation f.

- The computation is performed analogously:

$$h_i = f\left(\sum_j x_j w^{(h)}_{j,i} + b^{(h)}_i\right),$$
$$y_i = a\left(\sum_j h_j w^{(y)}_{j,i} + b^{(y)}_i\right),$$

or in matrix form

$$\boldsymbol{h} = f\left(\boldsymbol{x}^T\boldsymbol{W}^{(h)} + \boldsymbol{b}^{(h)}\right),$$
$$\boldsymbol{y} = a\left(\boldsymbol{h}^T\boldsymbol{W}^{(y)} + \boldsymbol{b}^{(y)}\right),$$



and for batch of inputs $\boldsymbol{H} = f\left(\boldsymbol{X}\boldsymbol{W}^{(h)} + \boldsymbol{b}^{(h)}\right)$ and $\boldsymbol{Y} = a\left(\boldsymbol{H}\boldsymbol{W}^{(y)} + \boldsymbol{b}^{(y)}\right)$.

- List the definitions of frequently used MLP output layer activations (the ones producing parameters of a Bernoulli distribution and a categorical distribution). Then write down three commonly used hidden layer activations (sigmoid, tanh, ReLU). **[10]**
    - **Bernoulli distribution**: $\sigma(x) = \frac{1}{1+e^{-x}}$, model predikuje pravděpodobnost
    - K**-kategorická distribuce**: $\text{softmax}(\boldsymbol{x}) \propto= e^{\boldsymbol{x}}$, $\text{softmax}(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$

- **sigmoid**: $\sigma(x) = \frac{1}{1+e^{-x}}$
- **tanh**: $2\sigma(2x) - 1$ (vychází ze sigmoidy, symetrická a derivace v nule je nula)
- **ReLU**: $\max(0, x)$

Questions@:, Lecture 5 Questions

- Considering a single-layer MLP with D input neurons, a ReLU hidden layer with H units and a softmax output layer with K units, write down the explicit formulas of the gradient of all the MLP parameters (two weight matrices and two bias vectors), assuming input $\boldsymbol{x}$, target t, and negative log likelihood loss. **[20]**
  - TODO:
- Formulate the Universal approximation theorem. **[10]**
  - nechť $\phi(x) : \mathbb{R} \to \mathbb{R}$ je nekonstantní, omezená a neklesající spojitá funkce, rozmačkávací funkce (např. ReLU)
  - pak pro všechna $\epsilon > 0$ a všechny spojité funkce $f : [0,1]^D \to \mathbb{R}$ existuje $H \in \mathbb{N}$, $\boldsymbol{v} \in \mathbb{R}^H$,$\boldsymbol{b} \in \mathbb{R}^H$ a $\boldsymbol{W} \in \mathbb{R}^{D \times H}$ takové, že označíme-li:

$$F(\boldsymbol{x}) = \boldsymbol{v}^T \phi(\boldsymbol{x}^T \boldsymbol{W} + \boldsymbol{b}) = \sum_{i=1}^{H} v_i \phi(\boldsymbol{x}^T \boldsymbol{W}_{(*,i)} + b_i)$$

  - (ϕ se aplikuje postupně na jednotlivé prvky)
  - pak platí pro všechny $\boldsymbol{x} \in [0,1]^D$:

$$|F(\boldsymbol{x}) - f(\boldsymbol{x})| < \epsilon$$

  - ve zkratce: funkce z podmínek se dá lineárně aproximovat pomocí nějakých vah $\boldsymbol{W}$, biasu $\boldsymbol{b}$ a "aktivační funkcí" $\phi(\boldsymbol{x})$, nebo tak něco
  - důkaz, že mlp je super model, aneb že umí aproximovat libovolnou spojitou funkci, s chybovostí až epsilon, tj. že existujou parametry H, v, b, W pro mlp
- How do we search for a minimum of a function $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$ subject to equality constraints $g_1(\boldsymbol{x}) = 0, \ldots, g_m(\boldsymbol{x}) = 0$? **[10]**
  - použijeme Lagrangeovy multiplikátory
  - nechť $f(\boldsymbol{x})$ je diferencovatelná a $\boldsymbol{\lambda} \in \mathbb{R}^m$
  - a množina derviací všech $\boldsymbol{g}(\boldsymbol{x})$ je lineárně nezávislá
  - Lagrangeova funkce: $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + < \boldsymbol{\lambda}, \boldsymbol{g}(\boldsymbol{x}) >$
  - řešímě soustavu parciálních derivací $\mathcal{L}$ rovných nule podle všech možných proměnných a $\boldsymbol{g}(\boldsymbol{x}) = 0$:
  - $\forall x \in \boldsymbol{x} : \frac{\partial \mathcal{L}}{\partial x} = 0 \wedge \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$
  - minimum bude alespoň jeden z bodů, které vyjdou z této soustavy rovnic
- Prove which categorical distribution with N classes has maximum entropy. **[10]**
  - TODO: prezentace slide 16

- Consider derivation of softmax using maximum entropy principle, assuming we have a dataset of N examples $(x_i, t_i), x_i \in \mathbb{R}^D, t_i \in \{1, 2, \ldots, K\}$. Formulate the three conditions we impose on the searched $\pi : \mathbb{R}^D \to \mathbb{R}^K$, and write down the Lagrangian to be minimized. **[20]**
    - TODO:
- Define precision (including true positives and others), recall, F_1 score, and F_β score (we stated several formulations for F_1 and F_β scores; any one of them will do). **[10]**
    - vždycky to je "trefil jsem se?" "co jsem predikoval"
	Target positive	**Target negative**
Predicted positive	True Positive (TP)	False Positive (FP)
Predicted negative	False Negative (FN)	True Negative (TN)
    - $$\text{precision} = \frac{TP}{TP + FP}$$
    - $$\text{recall} = \frac{TP}{TP + FN}$$
    - $$F_1 = \frac{TP + TP}{TP + FP + TP + FN}$$
    - $$F_\beta = \frac{TP + \beta^2 \cdot TP}{TP + FP + \beta^2 \cdot (TP + FN)}$$
- Explain the difference between micro-averaged and macro-averaged F_1 scores. **[10]**
    - micro-averaged F_1 score: sečtou se všechna TP, FP a FN přes všechny *binary classifications* a z těch se spočte F_1-skóre (četnost jednotlivých class je zohledněna)
    - macro-averaged F_1 score: pro každou *binary classification* se F_1-skóre spočte zvlášť a výsledkem je jejich průměr (četnost je ignorována)
- Explain (using examples) why accuracy is not a suitable metric for unbalanced target classes, e.g., for a diagnostic test for a contagious desease. **[5]**
    - Když 99 % populace nebude trpět danou nemocí a náš model pro všechy možné vstupy odpoví "je zdráv jako řípa", pak budeme mít přesnost (accuracy) 99 % a krev milionů lidí na svých rukou. Je sice super, že máme tak přesný model, ale byli bychom mnohem radši, kdybychom nemoc dokázali odhalit.
    - jop, jop chceme nějak i počítat úspěšnost u tříd s malou pravěpodobností, např. model má 0% úspěšnost, když týpek je nemocný, protože je biased a naučil se z dat, že prostě říct, že je zdravý, je big brain move.

Questions@:, Lecture 6 Questions

- Explain how is the TF-IDF weight of a given document-term pair computed. **[5]**
    - **binární indikátory**: pojem se v dokumentu vyskytuje, $1/0$
    - **term frequency (TF)**: relativní výskyt pojmu v dokumentu

- $$TF(t;d) = \frac{\text{počet výskytů pojmů } t \text{ v dokumentu } d}{\text{celkový počet pojmů v dokumentu } d}$$
    - **inverse document frequency (IDF)**: pojem můžeme reprezentovat jako *self-information*
    - $$IDF(t) = \log\left(\frac{\text{počet dokumentů}}{\text{počet dokumentů, kde se } t \text{ vyskytuje alespoň jednou}}\right)$$
    - tady se to dá krásně zdůvodnit: mějme nějaký obrovský počet dokumentů, pokud se pojem vyskytuje jenom v jednotkách, pak nám vyjde $\log(\text{velké číslo}) \to$ významnost tohoto pojmu vzhledem k dokumentům je velká
    - TF-IDF se počítá jako $TF \times IDF$, vyjadřuje to jak moc je pojem důležitý pro dokument v souboru dokumentů

- Define conditional entropy, mutual information, write down the relation between them, and finally prove that mutual information is zero if and only if the two random variables are independent (you do not need to prove statements about D_{KL}). **[10]**
    - **conditional entropy**:
    - $$H(Y|X) = \mathbb{E}_{x,y}[I(y|x)] = -\sum_{x,y} P(x,y)\log P(y|x)$$
    - **mutual information**:
    - $$I(X;Y) = \mathbb{E}_{x,y}\left[\log\frac{P(x,y)}{P(x)P(y)}\right]$$
    - **relation**:
    - $$H(Y) - H(Y|X) = I(X;Y)$$
    - mutual information je symetrická:
    - $$I(X;Y) = I(Y;X)$$
    - **důkaz**:
    - $$I(X;Y) = D_{KL}\left(P(X,Y)||P(X)P(Y)\right)$$
    - $$I(X;Y) \geq 0$$
    - $$I(X;Y) = 0 \iff P(X,Y) = P(X)P(Y) \iff \text{náhodné proměnné jsou nezávislé}$$

- Show that TF-IDF terms can be considered portions of suitable mutual information. **[10]**
    - prezentace 9

- Explain the concept of word embedding in the context of MLP and how it relates to representation learning. **[5]**
    - via Lukáš:
    - repre learning je o to, že si předtrénujeme reusable features pro další modely
    - v kontextu MLP: původní model byl language model, bral one hot encoded tokeny a násobil jimi matici vah, tím pádem v konkrétní vektor z matice je word embedding pro dané slovo
    - předpokládejme omezený slovník, pak slovo může být reprezentováno jako one-hot vektor

    - matice $\times$ one-hot slovo -> výběr jednoho vektoru z matice (one-hot je $(0, ..., 0, 1, 0, ..., 0)$), tenhle konkrétní výběr je **word embedding**
    - repre learning je něco, kde můžeme používat prosxy task, který vede k znovupoužitelným features; pro language modely jse embedding matrix používaný znovu a znovu (tedy opakovaně) pro všechna vstupní slova

- Describe the skip-gram model trained using negative sampling. **[10]**

    - mám okénko procházím jím text a snažím se naučit slovo, které je uprostřed (na řadě)
    - pro každé lsovo z jeho okolí vytvoříme trénovací data
    - A! přidáme ještě negativně (negative sampling) ohodnocená slova, která nasamplujeme z celého testu (může se stát, že se trefíme do slov, která s konkrétním slovem souvisí, ovšem praděpobnost je nízká)
    - Lenka dodává:
    - v tomhle jsem docela ztracená, jukla jsem na část přednášky, ale moc mi to nepomohlo
    - skip-gram model - vezmu slovo z prostředního okénka a snažím se předpovědět slova napravo a nalevo od něj
    - **negative sampling**:
        - pro každé slovo se chceme naučit pravděpodobnost, že j-té slovo je v okolí i-tého slova
        - naučíme se V^2 klasifikátorů logistických regresí (což je dost hodně, ale logistické regrese sdílejí parametry, takže to není zas až tak tragický)
        - loss function $-\log \sigma(\boldsymbol{e}_w^T \boldsymbol{v}_c)$, kde v_c je word embedding kontextového slova c a e_w je embedding vstupního slova w
        - problém ale je, že nemáme negativní příklady (v trénovacích datech máme jenom slova s jejich sousedními slovy, takže je sice super, že máme logistický regrese, ty se ale naučej, že mají vždycky říkat: "Joo, tohle je jistě soused, neexistuje nic jako nesoused.")
        - takže si negativní příklady dovyrobíme - nasamplujeme K negativních slovních příkladů c_i
        - takže k loss function ještě přidáme:
        - $$-\sum_{i=1}^{K} \log \sigma(-\boldsymbol{e}_w^T \boldsymbol{v}_{c_i})$$

- How would you proceed to train a part-of-speech tagger (i.e., you want to assign each word with its part of speech) if you only could use pre-trained word embeddings and MLP classifier? **[5]**

    - použijeme posouvací okénko embeddingů a klasifikuje prostřední slovo
    - (tohle spíše není dost rozepsané ke zkoušce)

Questions@:, Lecture 7 Questions

- Describe k-nearest neighbors prediction, both for regression and classification. Define L_p norm and describe uniform, inverse, and softmax weighting. **[10]**

    - predikce **regrese**:

    - $$t = \sum_i \frac{w_i}{\sum_j w_j} t_i$$
    - predikce **klasifikace**: je predikována nejčastější třída (`np.bincount`)
    - $$\boldsymbol{t} = \sum_i \frac{w_i}{\sum_j w_j} \boldsymbol{t}_i$$
    - $\boldsymbol{t}_i$ jsou distribuce v kategoriích, $\boldsymbol{t}_i \in \mathbb{R}^K$, kde K je počet tříd, se kterými pracujeme
    - L_p **norma**:
    - $$||x - y||_p = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$
    - standardně je to $1, 2, 3, \inf$
    - **uniform**: hlas každého z k sousedů má stejnou váhu
    - **inverse**: váha hlasu je úměrná 1/vzdálenost (hlas je dál, má nižší váhu)
    - **softmax**: váha hlasu je $\mathrm{softmax}$(- vzdálenost)
- Show that L^2-regularization can be obtained from a suitable prior by Bayesian inference (from the MAP estimate). **[10]**
    - TODO:
- Write down how $p(C_k|\boldsymbol{x})$ is approximated in a Naive Bayes classifier, explicitly state the Naive Bayes assumption, and show how is the prediction performed. **[10]**
    - TODO:
- Considering a Gaussian naive Bayes, describe how are $p(x_d|C_k)$ modeled (what distribution and which parameters does it have) and how we estimate it during fitting. **[10]**
    - TODO:
- Considering a Bernoulli naive Bayes, describe how are $p(x_d|C_k)$ modeled (what distribution and which parameters does it have) and how we estimate it during fitting. **[10]**
    - TODO:

Questions@:, Lecture 8 Questions

- Prove that independent discrete random variables are uncorrelated. **[10]**
    - Dvě na sobě nezávislé diskrétní náhodné proměnné nekorelují (to je strašná věta toto):
    - $$\begin{aligned} \mathrm{Cov}(x, y) &= \mathbb{E}\left[(x - \mathbb{E}[x])(y - \mathbb{E}[y])\right] \\ &= \sum_{x,y} P(x, y)(x - \mathbb{E}[x])(y - \mathbb{E}[y]) \\ &= \sum_{x,y} P(x)(x - \mathbb{E}[x]) P(y)(y - \mathbb{E}[y]) \\ &= \sum_{x} P(x)(x - \mathbb{E}[x]) \sum_{y} P(y)(y - \mathbb{E}[y]) \\ &= \mathbb{E}_x\left[x - \mathbb{E}[x]\right] \mathbb{E}_y\left[y - \mathbb{E}[y]\right] = 0 \ \square \end{aligned}$$

- Write down the definition of covariance and Pearson correlation coefficient ρ, including its range. **[10]**
    - **kovariance**:
    - $$\mathrm{Cov}(x, y) = \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])]$$
    - "střední hodnota součinu (x - střední hodnota)(y ...)"
    - $$\rho = \frac{\mathrm{Cov}(x, y)}{\sqrt{\mathrm{Var}(x)}\sqrt{\mathrm{Var}(y)}}$$
    - **rozsah** ρ je $[-1, 1]$
- Explain how are the Spearman's rank correlation coefficient and the Kendall rank correlation coefficient computed (no need to describe the Pearson correlation coefficient). **[10]**
    - **Spearman**: Spearmanovo ρ je Pearsonův korelační koeficient měřený na rancích původních dat, kde rank nějakého elementu je index tohoto prvku ve vzestupně seřazené posloupnosti (via Lukáš: mám data (x,y), seřadím jak X tak Y a koreluju jejich indexy seřazení)
    - **Kendall**: vygeneruje dvojice všechny x a všechny y, měří počet *concordant pairs* (páry, kde y klesá/stoupá, právě tehdy když x kleasá/stoupá) mínus *discordant pairs* (páry, kde y klesá/stoupá, právě tehdy když x stoupá/klesá (přesně naopak než minule)) a to je potřeba vydělit počtem párů, značí se τ
- Describe setups where a correlation coefficient might be a good evaluation metric. **[5]**
    - Lenka:
    - korelace se jako metrika používá v úlohách, kde:
        - měříme podobnost - např. word embeddings (Máme výsledky psycholingvistických průzkumů, jak lidem připadají jaká slova podobná. Chceme měřit vzdálenost mezi našimi word embeddings a chceme, aby tato naměřená vzdálenost korelovala s výsledky psycholingvistického průzkumu. Používá se Pearsonova nebo Spearmanova korelace.)
        - vyhledáváme a seřadíme výsledky - např. document retrieval (Seřazujeme od nejlepšího po nejhorší, používá se Kendallovo τ nebo Spearmanova korelace.)
- Describe under what circumstance correlation can be used to assess validity of evaluation metrics. **[5]**
    - kontrola gramatiky, pravopisu - koreluje náhled člověka s výsledky stroje
    - Lenka: pan Straka říká:
    - Lidi hodnotí, jak se jim líbí výsledky (např. hodnotí vygenerovanou řeč na stupnici 1-10, 1 znamená děs a hrůza, 10 znamená, že je to nerozlišitelné od člověka), ale pozor na lidi, kteří za toto hodnocení dostávají zaplaceno (drží enter, peníze naskakují, jejich hodnocení nám toho ale moc neřekne). Takže chceme vědět, jak moc se lidi shodují, když to hodnocení dělají. Zajímá nás tedy korelace. Ale teda nevim, jestli je to zrovna odpověď na tuto otázku...
- Define Cohen's κ and explain what is used for when preparing data for machine learning. **[10]**
    - $$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

    - *Takže víš, Cohenova Kappa je prostě takovej trik, co pomáhá vidět, jak moc se dva hodnotitelé shodujou v klasifikaci, přičemž bere v úvahu, že část té shody by mohla být čistě náhodná.* - via Chat GPT
    - p_0: je pozorovaná shoda
    - p_e: je očekávaná shoda, pravděpodobnost, že se hodnotitelé náhodně (omylem) trefí do stejného ohodnocení
    - v ML se používá např. k vyfiltrování *confusing data points* (lidi moc nevědí, jak to oanotovat, sporná data), díky Cohenově κ lze zjistit, že jsme anotátorům nedali jasné instrukce a oni nevěděli, co mají dělat - ale pozor, anotátoři s nízkou shodou nutně nemusí být lajdáci nebo blbečci, možná to jenom znamená jiné kulturní zázemí atd. a v takovém případě je asi úplně nechcem vyházet

- Considering an averaging ensemble of M models, prove the relation between the average mean squared error of the ensemble and the average error of the individual models, assuming the model errors have zero means and are uncorrelated. **[20]**

    - TODO:

- Explain knowledge distillation: what it is used for, describe how it is done. **[10]**

    - mám velký model, který toho hodně umí a vrací mi pravděpodbnosti jednotlivých class (předpovědi)
    - pak začnu trénovat nový model, který dostává vstupní data stejná jako velký model, ovšem targety jsou vyměněny za předpovědi velkého modelu
    - aka místo na skalárech trénuji model na vektorech pravděpodobností jednotlivých tříd
    - cílem je, aby se model zmenšil, případně i zrychlil
    - tady doporučuji nakreslit nějakou pěknou ilustraci celého procesu (pro bodíky ze soucitu)

Questions@:, Lecture 9 Questions

- In a regression decision tree, state what values are kept in internal nodes, define the squared error criterion and describe how is a leaf split during training (without discussing splitting constraints). **[10]**

    - v interních vrcholech je potřeba uchovávat trvale pouze "podle které featury se splituje" a jaká je splitovací hranice
    - $I_\mathcal{T}$ indexy dat, která patří do daného vrcholu
    - $\mathcal{T}$ je vrchol stromu, pak:
    - $$c_{\mathrm{SE}}(\mathcal{T}) = \sum_{i \in I_\mathcal{T}} (t_i - t_\mathcal{T})^2, \text{kde } t_\mathcal{T} = \frac{1}{|\mathcal{T}|} \sum_{i \in I_\mathcal{T}} t_i$$
    - (kvadratická odchylka od aritmetického průměru všech těch příslušných tagetů)
    - vrchol dělíme podle:
        1. featury (for cyklus přes všechny featury)
        2. její hodnoty (for cyklus přes všechny unikátní hodnoty featury)
    - minimalizujeme $c_{\mathcal{T}(L)} + c_{\mathcal{T}(R)} - c_\mathcal{T}$ (dítě 1 plus dítě 2 minus rodič (jejich criteriony))

- In a K-class classification decision tree, state what values are kept in internal nodes, define the Gini index and describe how is a node split during training (without discussing splitting constraints).

[10]

- internal je stejný jako u předchozího, TODO: překopírovat
- $p_{\mathcal{T}}(k)$ je průměrná pravděpodobnost třídy k ve vrcholu $\mathcal{T}$
- Gini:
- $$c_{\text{Gini}}(\mathcal{T}) = |I_{\mathcal{T}}| \sum_k p_{\mathcal{T}}(k)(1 - p_{\mathcal{T}}(k))$$
- splitování je stejné

- In a K-class classification decision tree, state what values are kept in internal nodes, define the entropy criterion and describe how is a node split during training (without discussing splitting constraints). **[10]**
  - všechno, co není zmíněno, je zmíněno výše. TODO: překopírovat
  - DONE: překopírovat entropy criterion str 8
  - Entropy criterion:
  - $$c_{\text{entropy}}(\mathcal{T}) = |I_{\mathcal{T}}| \cdot H(\boldsymbol{p}_{\mathcal{T}}) = -|I_{\mathcal{T}}| \sum_{k, p_{\mathcal{T}}(k) \neq 0} p_{\mathcal{T}}(k) \log p_{\mathcal{T}}(k)$$

- For binary classification, derive the Gini index from a squared error loss. **[20]**
  - tahle úloha je vlastně v pohodě
  - $x_i \in \{0, 1\}$
  - p je reprezentant v listu, neboli praavděpodobnost jedničky
  - $\#1$ počet jedniček v N případech
  - chceme minimalizovat square error:
  - $$\begin{aligned} \frac{\partial \sum_i (p - x_i)^2}{\partial p} &= 0 \\ \sum_i (p - x_i) &= 0 \\ \sum_i p - \sum_i x_i &= 0 \\ Np - \#1 &= 0 \\ (\#0 + \#1)p - \#1 &= 0 \\ \implies p &= \frac{\#1}{\#0 + \#1} \end{aligned}$$
  - teď dosadíme do SEL:
  - $$\sum \left(\frac{\#1}{\#0 + \#1} - x_i \right)^2 = \#0 \left(\frac{\#1}{\#0 + \#1} - 0 \right)^2 + \#1 \left(\frac{\#1}{\#0 + \#1} - 1 \right)^2 =$$
  - $$= \frac{\#0(\#1)^2}{(\#0 + \#1)^2} + \frac{\#1(\#0)^2}{(\#0 + \#1)^2} = \frac{\#0\#1(\#0 + \#1)}{(\#0 + \#1)^2}$$
  - $$= |\text{velikost množiny}| \frac{\#0}{(\#0 + \#1)} \frac{\#1}{(\#0 + \#1)} = |\mathcal{I}| p(1 - p)$$

- For K-class classification, derive the entropy criterion from a non-averaged NLL loss. **[20]**
  - TODO:

- Describe how is a random forest trained (including bagging and a random subset of features) and how is prediction performed for regression and classification. **[10]**
    - použijeme stejné stromy jako pro "decision tree"
    - strom trénujeme standardně jako "decision tree", ale:
    - **feature subsampling**: při hledání nejlepšího splitu se vyzkouší jen nějaká podmnožina všechn možných featur
    - **bagging**: každý strom se trénuje na jiném bootstrap samplu
    - **predikce**: pro jeden strom se dělá viz výše (pro regresi je to průměrná hodnota v listu, kam dojdeme; pro klasifikace je to nejčastější třída v listu), výsledky z jednotlivých stromů poté hlasují o finálním výsledku

Questions@:, Lecture 10 Questions

- Write down the loss function which we optimize in gradient boosted decision trees during the construction of t^{th} tree. Then define g_i and h_i and show the value $w_{\mathcal{T}}$ of optimal prediction in node $\mathcal{T}$ and the criterion used during node splitting. **[20]**
    - TODO:
- For a K-class classification, describe how to perform prediction with a gradient boosted decision tree trained for T time steps (how the individual trees perform prediction and how are the $K \cdot T$ trees combined to produce the predicted categorical distribution). **[10]**
    - TODO: prezentace str 16
- What type of data are gradient boosted decision trees good for as opposed to multilayer perceptron? Explain the intuition why it is the case. **[5]**
    - **MLP** (jedna, dvě skryté vrstvy): multidimenzionální data (obrázky, řeč, text), jednotlivá featura neznamená zhola nic
    - **GBDT**: data s nižšími dimenzemi, vstupní featury jsou lehce interpretovetelné

Questions@:, Lecture 11 Questions

- Formulate SVD decomposition of matrix $\boldsymbol{X}$, describe properties of individual parts of the decomposition. Explain what the reduced version of SVD is. **[10]**
    - **Vlastnosti jednotlivých prvků dekompozice**:
    - SVD = *Singular Value Decomposition*
    - Každá (i obdélníková) matice $\boldsymbol{X}$ dimenze $m \times n$ a ranku r může být vyjádřena jako:
    - $$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$
    - $\boldsymbol{U}$ je $m \times m$ orthonormální matice
    - $\boldsymbol{\Sigma}$ je $m \times n$ diagonální matice s nezápornými prvky, takzvanými *singular values*, které jsou seřazené sestupně

- $\boldsymbol{V}$ je $n \times n$ orthonormální matice
- $$\boldsymbol{XV} = \boldsymbol{U\Sigma} \quad \Rightarrow \quad \boldsymbol{X}\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k \quad \forall k = 1, \ldots, r$$
- $$\boldsymbol{X} \begin{bmatrix} \vdots \\ \boldsymbol{v}_1 \cdots \boldsymbol{v}_r \cdots \boldsymbol{v}_n \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \boldsymbol{u}_1 \cdots \boldsymbol{u}_r \cdots \boldsymbol{u}_m \\ \vdots \end{bmatrix} \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \boldsymbol{0} \\ & & \sigma_r & \\ \hline & \boldsymbol{0} & & 0 \end{array}\right]$$
- **Redukovaná verze SVD**:
- Redukovaná verze **SVD**: Můžeme zahodit σ_k pro $k > r$ a díky tomu použít menší $\boldsymbol{U}$ a $\boldsymbol{V}$
- σ jsou seřazeny sestupně $\Rightarrow$ $\boldsymbol{X}$ můžeme aproximovat použitím $k < \min(m, n)$ (k je menší než menší rozměr)
- $$\tilde{\boldsymbol{X}} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$
- **Eckart-Young theorem** nám říká, že toto je nejlepší možná aproximace pro rank k vzhledem k Frobeniově normě.

- Formulate the Eckart-Young theorem. **[10]**
  - **Eckart-Young Theorem**:
  - $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ a $\boldsymbol{X}_k = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \ldots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$ je aproximace s pomocí **SVD**. Pak pro každé $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ ranku k platí:
  - $$||\boldsymbol{X} - \boldsymbol{X}_k||_F \leq ||\boldsymbol{X} - \boldsymbol{B}||_F.$$
  - ta F jsou, nejspíš, Frobeniovy normy
- Explain how to compute the PCA of dimension M using the SVD decomposition of a data matrix $\boldsymbol{X}$, and why it works. **[10]**
  - **PCA** je prvních M vetorů (sloupců) matice $\boldsymbol{V}$ s **SVD** $(1/N(\boldsymbol{X} - \text{mean}(x))^T(\boldsymbol{X} - \text{mean}(x)))$
- Given a data matrix $\boldsymbol{X}$, write down the algorithm for computing the PCA of dimension M using the power iteration algorithm. **[20]**

  Input: Real symmetric matrix $\boldsymbol{A}$ with a dominant eigenvalue.
  Output: The dominant eigenvalue λ and the corresponding eigenvector $\boldsymbol{v}$, with probability close to 1.
  - Initialize v randomly (for example each component from $U[-1, 1]$).
  - Repeat until convergence (or for a fixed number of iterations):
    - $\boldsymbol{v} \leftarrow \boldsymbol{Av}$
    - $\lambda \leftarrow \|\boldsymbol{v}\|$
    - $\boldsymbol{v} \leftarrow \boldsymbol{v}/\lambda$
  - If the algorithm converges, then $\boldsymbol{v} = \boldsymbol{Av}/\lambda$, so $\boldsymbol{v}$ is an eigenvector with eigenvalue λ.
- Describe the K-means algorithm, including the `kmeans++` initialization. **[20]**
  - TODO:

Questions@:, Lecture 12 Questions

- Considering statistical hypothesis testing, define type I errors and type II errors (in terms of the null hypothesis). Finally, define what a significance level is. **[10]**
    - H_0 je nulová hypotéza
    - **type I error**: false positive (zamítnutí H_0, přestože platí)
    - **type II error**: false negative (nezamítnutí H_0, přestože neplatí)
    - **hladina významnosti** je pravděpodobost chyby prvního druhu (aka, že se se dopouštíme chyby prvního druhu, když vyvozujeme výsledek); označuje se α
- Explain what a test statistic and a p-value are. **[10]**
    - via dlouhá session s Lukášem:
    - test statistic hodnota, díky které rozlišujeme nulovou a alternativní hypotézu a jsme schopni spočítat její distribuci
    - p-value je pravděpodobnost, že platí naše naměřená hodnota v distribuci nulové hypotézy nebo hodnota extrémnější
    - stare:
    - test statistic je shrnutí, soubor pozorovaných dat, velmi často se jedná o jednu hodnotu (střední hodnota), pomocí které můžeme rozlišit nulovou a alternativní hypotézu
    - p-value je pravděpodobnost, že naměříme test statistic alespoň tak extrémní jako to, které jsme pozorovali (jestliže platí H_0); nízké p-value znamená, že pozorovaná data jsou velmi nepravděpodobná pod H_0
- Write down the steps of a statistical hypothesis test, including a definition of a p-value. **[10]**
    1. Formulujeme nulovou hypotézu H_0 a můžeme formulovat i alternativní hypotézu H_1
    2. vybereme *test statistic*
    3. spočteme pozorovanou hodnotu *test statistic*
    4. spočteme p-value pro nulovou hypotézu H_0
    5. odmítneme nulovou hypotézu H_0 (případně ve prospěch H_1), jestliže p-value je pod vybranou hladinou významnosti α (**bonus**: nejčastější α jsou 5 %, 1 %, 0,5 % a 0,1 %)
- Explain the differences between a one-sample test, two-sample test, and a paired test. **[10]**
    - **one-sample** test: samplujeme z jednoho distribuce
    - **two-sample**: samplujeme ze dvou distribuce
    - **paired-test**: samplujeme ze dvou distribucí, ale samply jsou spárovány (často se spočítá rozdíl mezi spárovanými hodnotami a udělá se one-sample na střední hodnotě rozdílu)
    - poznatek po zkoušce: toto není dost ke zkoušce, informace zde uvedené jsou za 3/10
- When considering multiple comparison problem, define the family-wise error rate, and prove the Bonferroni correction, which allows limiting the family-wise error rate by a given α. **[10]**
    - **FWER** je pravděpodobnost, že nastane alespoň jedna chyba typu I ve famílii
    - $$\mathrm{FWER} = P\left(\bigcup_i (p_i \leq \alpha)\right)$$

    - FWER se dát limitovat pomocí α pomocí **Bonferri correction**, která odmítá H_0 testování famílie o velikost m, pokud $p_i < \frac{\alpha}{m}$:
    - Předpokládejme takovou úpravu a použijme **Boolovu nerovnost**:
    - **Boolova nerovnost**:
    - $$P\left(\bigcup_i A_i\right) \leq \sum_i P(A_i)$$
    - z toho máme:
    - $$\text{FWER} = P\left(\bigcup_i (p_i \leq \frac{\alpha}{m})\right) \leq \sum_i P\left(p_i \leq \frac{\alpha}{m}\right) = m \cdot \frac{\alpha}{m} = \alpha$$
    - což ukazuje to, že se FWER dá omezit α (přesně to, co jsme chtěli)
    - (mnemotechnické pomůcky: FWER je jako horečka anglicky a PUpi je skoro jako štěnátko anglicky)

- For a trained model and a given test set with N examples and metric E, write how to estimate 95% confidence intervals using bootstrap resampling. **[10]**
    - TODO:

- For two trained models and a given test set with N examples and metric E, explain how to perform a paired bootstrap test that the first model is better than the other. **[10]**
    - TODO:

- For two trained models and a given test set with N examples and metric E, explain how to perform a random permutation test that the first model is better than the other with a significance level α. **[10]**
    - TODO:

Questions@:, Lecture 13 Questions

- Explain the difference between deontological and utilitarian ethics. List examples on how these theoretical frameworks can be applied in machine learning ethics. **[10]**
    - **deontologie** je etika založená na principech a na pravidlech, snažíme se vymyslet dobrá pravidla a dobré principy, dobrý skutek je ten, který dodržuje tato dobrá pravidla a dobré principy (není ani tak důležité, jak skutek dopadl, hlavně že to ten člověk myslel dobře a dodržoval dobrá pravidla)
    - **deontologie v ML** používání dobrých principů jako jsou nediskriminace, autonomie + princip informovaného souhlasu, soukromí, *beneficient*, vůbec se nemluví o důsledcích, nikdy nepoužívat rasu lidí jako featuru
    - **utilitarismus** je etika, která říká, že důležité jsou důsledky skutků (tj. dobrý skutek je takový skutek, který má dobré následky, je jedno, jak jsme to mysleli a jestli jsme dodržovali nějaká dobrá pravidla), cílem je maximalizovat štěstí na světě a minimalizovat škodu, nevýhodou je, že nehledíme na práva jednotlivců, jde nám o celou společnost
    - **utilitarismus v ML** kdo bude technologií ovlivněn a jak? Může to způsobit nějakou újmu (kvůli této technologii zlý extrémista vyhraje volby, tato technologie způsobí šílenou

uhlíkovou stopu...), zamýšlíme se nad tím, jak se těmto ošklivým důsledkům vyhnout

- List few examples of potential ethical problems related to data collection. **[5]**
    - data mohou být nereprezentativní (chybí minority, chudí lidé, overreprezentované západní země - pro nezápadní země nemusí technologie fungovat správně...)
    - historical bias - něco, co platilo v minulosti, už nemusí platit nyní
    - problémy s copyrightem
    - problematické sbírání dat
        - **crowdsourcing** - lidi jsou najímáni, aby dělali to, co bude dělat ML model, často špatně placená práce, často v zemích třetího světa, špatné důsledky pro psychiku (koukání na fuj obrázky)
        - trénovací data jsou sbírána od uživatelů, kteří nemají moc jinou možnost, než s tím souhlasit

- List few examples of potential ethical problems that can originate in model evaluation. **[5]**
    - problém s diskrétními výstupy (máme 49 % třída jedna, 51 % třída druhá, rozhodneme se pro druhou třídu a hotovo)
    - vybereme si nevhodnou evaluační metriku - může se stát, že metrika říká "To je super, jsi šikulák!", ale úplně vyignoruje např. gender bias nebo něco může fungovat hooodně blbě pro specifické skupiny lidí (nářečí atd.)
    - vybírání zaměstnanců na základě jejich CV za pomocí precision, ale úplně ignorujeme recall (nikdo se neptá, jestli jsm doporučili všechny dobré kandidáty, stačí, že jsme doporučili dobré kandidáty, takže model má tendence nenavrhovat lidi s romskym příjmením, je to rasista atd.)