Билет 1

Модель нейрона Маккалока-Питтса. Персептрон Розенблатта. Теорема Новикова. Полносвязные нейронные сети

1) $X=(x_1, \dots x_n)$, нейрон $a(x)$ вычисляет n-арную булеву функцию: $a(x) = \text{Heaviside}(\sum_{j=1}^{n} w_j x^j - w_0)$, w - веса ($w_j>0$ - возб., иначе - тормозящий)

2) Виды элементов: S- сенсорный эл-т, A - ассоциативный эл-т, ему соотв. несколько S- элементов. A активируется, если количество сигналов S на его входе превысило не-которое значение Θ. Затем сигналы ~~идут~~ от возбужденных A идут на сумматор R с коэф. w_j - весом A-R связи (веса S-A принимают знач -1, 0 или 1) значения порогов A рандомны и неизменны, общ. вид ф-ции, реализуемой R-эл-том, представим как $\text{perceptron}(x) = \text{sgn}(\sum_{j=1}^{n} w_j x_j - \Theta)$

3) Th Новикова и правило Хэбба: Пусть мн-во пре-диктов $Y=\{-1,+1\}$, x - объект обуч. выборки $X^\ell = \{x_j, y_j\}_{j=1}^{\ell}$, $y_j = y_j^*(x_j) \in Y$ - класс. Алг. классиф. имеет вид $a(x,w) = \text{sgn}(\langle x, w\rangle)$, ошибка видимая если знак $\langle , \rangle$ не совпал со знаком класса, т.е. $\langle x\, w\rangle y<0$. Можно модифицировать веса: $w \to w + \eta x y$

Собственно теорема: $X = \mathbb{R}^{n+1}$, $Y=\{-1, 1\}$, X^ℓ (выборка) лин. разделима, т.е. $\exists w^*$ и $\delta>0$ т.ч. $\langle x_j, w^*\rangle y_j > \delta$ $\forall j=1, \dots \ell$. Тогда алгоритм может найти вектор весов, разд. train без ошибок за кон. число итераций $\forall$ нач. прибл. w_0 и $\forall \eta>0$.

* FCN: input $\to x \to xw+b \to \sigma \to$ output

dense layer - лин. преобр. входн. данных (обуч. параметры матрица w и вектор b): $x \mapsto xW+b$, $W \in \mathbb{R}^{d\times k}$, $x\in\mathbb{R}^d$, $b\in\mathbb{R}^k$. Слой делает из d-мерных векторов k-мерные.

activation - нелин. преобр.

Билет 2

Аппрокс. теоремы (БД): Лузина, Колмогорова-Арнольда, Цыбенко

1) Лузина: $\forall$ измеримой f на $[a, b]$ и $\forall \varepsilon>0$ $\exists f_\varepsilon \in C[a,b]$, такая что $f=f_\varepsilon$ везде кроме некоторого мн-ва меры ε.

Т.е. $\mu\{x\in[a,b]: f(x)\neq f_\varepsilon(x)\}<\varepsilon$ стр. 2

② Теорема Колмогорова - Арнольда: всякая непр. ф-ция представима в виде композиции непр. ф-ций одной пер.

Т.е. $\forall f(x)$, $x=(x_1,\dots x_n)$ $\exists$ представл. $\sum_{q=0}^{2n}\Phi_q\left(\sum_{p=1}^{n}\varphi_{q,p}(x_p)\right)$

Усиление Шпрехера: $f=\sum_{q=0}^{2n}\Phi\left(\sum_{p=1}^{n}\lambda_p\varphi(x_p+\eta q)+q\right)$, $\eta,\lambda\in\mathbb{R}$, $\Phi\in C$,
$\Phi: R\to R$, $\varphi[0,2]\to R$.

③ Цыбенко: $\forall f\in C^0(R)$, $f: R\to R$, $\forall\varepsilon>0\ \exists N, w_1\dots w_N, b_1\dots b_N,$
$\alpha_1\dots\alpha_N: \left|f-\sum_{i=1}^{N}\alpha_i\sigma(\langle x,w_i\rangle+b_i)\right|<\varepsilon\ \ \forall x\in[0,1]^m\in\mathbb{R}^m$

Билет 3

Обучение искусств. нейронов. stochastic gradient descent, back-propagation.

Минимизируем: $Q(w)=\sum h(w,x_i,y_i)\to\min$

$w=w-\eta h_i(w)$, $\nabla h_i(w)=\left(\frac{\delta h_i(w^k)}{\delta w^k}\right)_{k=1}^{m}$, $Q=(1-\lambda)Q+\lambda h_i(w)$

Backdrop

дано $(\{x_i,y_i\})_{i=1}^{\ell}$, сети $(H_\ell)_{\ell=1}^{h}$, парам. η,λ. Ищем вектор весов всех слоев $w=(w^1\dots w^n)$

1) random $(x_i)\in X^\ell$, $\ell=1\dots h$, $n=1\dots H_\ell$

forward: $X^\ell_{in}=\sigma^\ell_n\left(\sum_{k=0}^{H_{\ell-1}}w^i_{kn}x^{\ell-1}_{in}\right)$ $\quad p^h_{ni}=\frac{\partial h_i(w)}{\partial x^w_n}$

$z^\ell_{in}=(\sigma^\ell_n)'\left(\sum_{k=0}^{H_{\ell-1}}w^i_{kn}x^{\ell-1}_{ik}\right)$

backward: $p^{\ell-1}_{ik}=\sum_{h=0}^{H_\ell}p^\ell_{in}z^\ell_{in}w^\ell_{kn}$

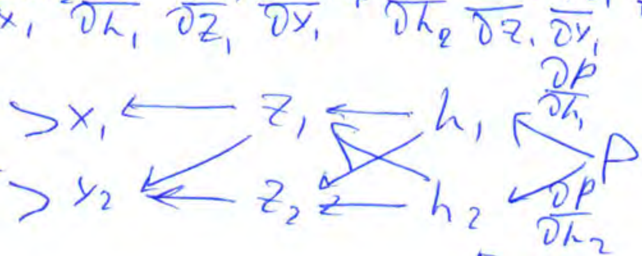
grad step for all $\ell=1\dots h$, $k=0\dots H_{\ell-1}$, $n=1\dots H_\ell$

$w^\ell_{kn}=w^\ell_{kn}-\eta p^\ell_{in}z^\ell_{in}x^{\ell-1}_{in}$

Пока Q и/или весов unstable

Пример: ③ $\frac{\partial P}{\partial h_1}$, $\frac{\partial P}{\partial h_2}$, ② $\frac{\partial P}{\partial z_i}=\frac{\partial P}{\partial h_1}\frac{\partial h}{\partial z_1}=\frac{\partial P}{\partial h_1}\frac{\partial h_2}{\partial z_1}$; $\frac{\partial P}{\partial z_2}=\frac{\partial P}{\partial h_1}\frac{dh_1}{\partial z_2}+\frac{\partial P}{\partial h_2}\frac{\partial h_2}{\partial z_2}$

① $\frac{\partial P}{\partial x_1}=\frac{\partial P}{\partial h_1}\frac{\partial h_1}{\partial z_1}\frac{\partial z_1}{\partial x_1}+\frac{\partial P}{\partial h_2}\frac{\partial h_2}{\partial z_1}\frac{\partial z_1}{\partial x_1}+\frac{\partial P}{\partial h_1}\frac{\partial h_1}{\partial z_2}\frac{\partial z_2}{\partial x_1}+\frac{\partial P}{\partial h_2}\frac{\partial h_2}{\partial z_2}\frac{\partial z_2}{\partial x_1}$



Билет 4

Метод обучения Adaptive momentum. Метод Drop-out и его реализация

① $v_0=0$

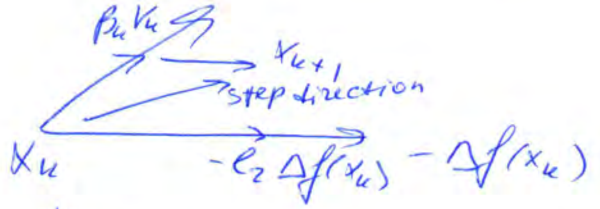
$v_{k+1}=\beta_1v_k+(1-\beta_1)\nabla f(x_k)$

$G_{k+1}=\beta_2G_k+(1-\beta_2)(\nabla f(x_k))^2$

$x_{k+1}=x_k-\frac{\ell_2}{\sqrt{G_{k+1}+\varepsilon}}v_{k+1}$

$\beta_1=0.9$
$\beta_2=0.99$
$\varepsilon=1e-8$
$f(x)=\sum h(x,y_i)$

Получим из momentum и RMS prop. (Слайд 3)



RMS Prop: $G_{k+1} = \gamma G_k + (1-\gamma)(\nabla f(x_k)^2)$

$x_{k+1} = x_k - \frac{\ell_2}{\sqrt{G_{k+1}+\varepsilon}} \nabla f(x_k)$

dropout

При град. шаге $h_i(w) \to \min\limits_w$ отключаем n-ый нейрон ℓ-го слоя с вер. p_ℓ

$x_{n_i}^{\ell} = \xi_n^{\ell}\, \sigma_n^{\ell}(\Sigma w_{kn}^{\ell} x_{k_i}^{\ell-1})$ $P(\xi_n^{\ell}=0) = p_\ell$

вклю. с $(1-p_\ell)$

на обуч. $x_{n_i}^{\ell} = \frac{1}{1-p_\ell} \xi_n^{\ell} \sigma_n^{\ell}(\Sigma w_{kn}^{\ell} x_{k_i}^{\ell-1})$

на валидации $x_{n_i}^{\ell} = \sigma_n^{\ell}(\Sigma w_{kn}^{\ell} x_{k_i}^{\ell-1})$

с учетом L_2 gradstep $w = w(1-\eta\lambda) - \eta \frac{1}{1-p_\ell} \xi_n^{\ell} h_i'(w)$

Билет 5

Основные слои искусств. нейросети:
свертки, skip-connection, drop-out, batch norm, нелинейности.

1) даны 2 матр. $A(n_x \times n_y)$ и $B(m_x \times m_y)$ $C = A * B$. C.shape = $(n_x - m_x + 1), (n_y - m_y + 1)$, т.е. $C_{i,j} = \sum\limits_{u=0}^{m_x-1} \sum\limits_{v=0}^{m_y-1} A_{i+u, j+v} B_{u,v}$

2) Skip-connection решает проблему затухания grad, передача инф. из нижних слоев в верхние:

$\to$ layer 1 $\to$ activation $\to$ layer 2 $\xrightarrow{\text{skip-connection}}$ $\oplus \to$ ..

3) dropout - случайное выключение некоторого числа нейронов для борьбы с переобучением и улучшения обобщающей способности

4) Batch norm - снимает зависимость от входн. данных $\mu = 0$, $\sigma = 1$
нелинейными слоями данные данные с

Batch $= \{x_1 \dots x_m\}$ $\mu_B = \frac{1}{m} \Sigma x_i$ $\sigma_B^2 = \frac{1}{m} \Sigma (x_i - \mu_B)^2$

$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$ $y_i = \gamma \hat{x}_i + \beta = \text{BatchNorm}(x_i)$ - отмасшт. и сдвиг.

5) ~~nonline~~ нелинейность помогает моделировать сложные ф-ции.

Рекуррентные нейронные сети. LSTM.

RNN-сети с циклами, подходящие для обработки последовательностей. Для обучения применяется BPTT-backprop through time. осущ. он так:

Схема слоя RNN и развернутая схема.

слои:

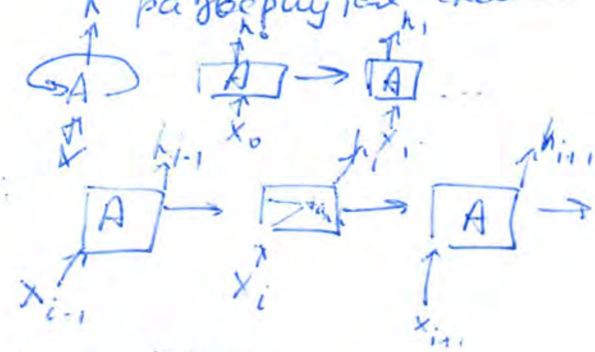
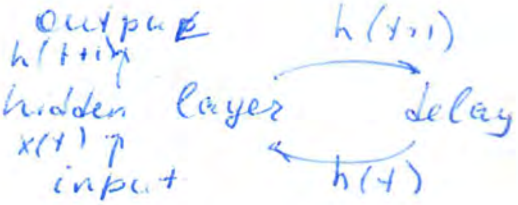


Схема с задержкой в скрытом слое:



Виды RNN:

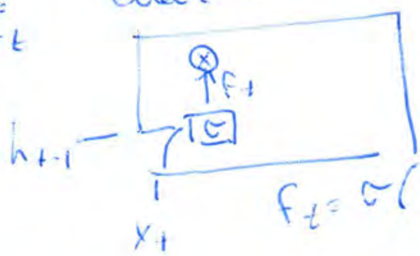
1) 1 вход, много выходов: для генерации аудио
2) Много входов, 1 выход: для оценки тональностей
3) много входов и выходов: для перевода

LSTM

помнит данные долгое время

шаг 1

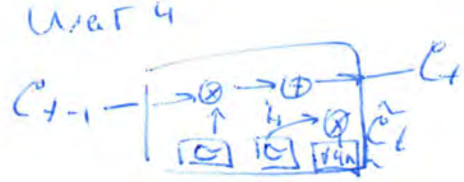
шаг 2



$f_t = \sigma(W_f(h_{t-1}, x_t) + b_f)$

шаг 3

шаг 4



$C_t = f_t C_{t-1} + i_t \tilde{C}_t$

$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i)$

$\tilde{C}_t = tgh(W_C [h_{t-1}, x_t] + b_C)$

$h_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \, tgh(C_t)$

Билет 7

Метод опорных векторов. Kernel trick.

Ставим задачу регрессии - дано train $X = \{(x_i, y_i)\}_{i=1}^{\ell}$
ищем $g(x)$ - аппроксимацию выборки лучшим образом.

loss: $h(y, g(x)) = \begin{cases} 0, & |y-g(x)| \le \varepsilon \\ |y-g(x)|-\varepsilon, & |y-g(x)| > \varepsilon \end{cases}, \quad \varepsilon > 0$

ищем решение в линейном виде $f(x) = (w, x) - w_0$

loss: $a(x_i): |(w, x_i) - w_0 - y_i|_\varepsilon \; \forall (x_i, y_i)$ где $|z|_\varepsilon = \max(0, |z| - \varepsilon)$

функционал потерь

$Q_\varepsilon(a, X) = \sum_{i=1}^{\ell} |(w, x_i) - w_0 - y_i|_\varepsilon + \tau (w, w)^2 \to \min_{w, w_0}$

введем переменные ξ_i^+ и ξ_i^- - знач. которых = потери при завышенном или заниженном ответе.

$\xi_i^+ = (a(x_i) - y_i - \varepsilon)_+ \quad \xi_i^- = (-a(x_i) + y_i - \varepsilon)_- \quad i = 1, \dots, \ell$

Задача минимизации:

$$\begin{cases} 0{,}5 (w, w)^2 + \frac{1}{2\tau} \sum_{i=1}^{\ell} (\xi_i^+ + \xi_i^-) \to \min\limits_{w, w_0, \xi_i^+, \xi_i^-} \\ (w, x_i) - w_0 \le y_i + \varepsilon + \xi_i^+ \\ (w, x_i) - w_0 \ge y_i - \varepsilon - \xi_i^- \\ \xi_i^- \ge 0, \; \xi_i^+ \ge 0 \end{cases}$$

Будем решать двойственную задачу, скал. произв. заменим ядром $k(x_i, x_j)$, λ_i^+, λ_i^- - двойств. перем.

$$\begin{cases} h(\lambda^+, \lambda^-) = -\varepsilon \sum_{i=1}^{\ell} (\lambda_i^+ + \lambda_i^-) + \sum_{i=1}^{\ell} (\lambda_i^- - \lambda_i^+) y_i - \frac{1}{2} \sum (\lambda_i^- - \lambda_i^+)(\lambda_j^- - \lambda_j^+) \cdot k(x_i, x_j) \to \max\limits_{\lambda^+, \lambda^-} \\ 0 \le \lambda_i^+ \le C = \frac{1}{2\tau} \quad 0 \le \lambda_i^- \le C \\ \sum (\lambda_i^+ + \lambda_i^-) \to 0 \end{cases}$$

В рез. все x_i делятся на 5 типов

1) $|a(x_i) - y_i| < \varepsilon \quad \lambda_i^+ = \lambda_i^- = \xi_i^+ = \xi_i^- = 0$
2) $a(x_i) = y_i + \varepsilon \quad 0 < \lambda_i^+ < C, \quad \lambda_i^- = \xi_i^+ = \xi_i^- = 0$
3) $a(x_i) = y_i + \varepsilon \quad 0 < \lambda_i^- < C, \quad \lambda_i^+ = \xi_i^+ = \xi_i^- = 0$
4) $a(x_i) > y_i + \varepsilon \quad \lambda_i^+ = C, \; \lambda_i^- = 0, \; \xi_i^+ = a(x_i) - y_i - \varepsilon, \; \xi_i^- = 0$
5) $a(x_i) < y_i - \varepsilon \quad \lambda_i^+ = 0, \; \lambda_i^- = C, \; \xi_i^+ = 0, \; \xi_i^- = y_i - a(x_i) - \varepsilon$

2-5 - опорные, уч. в опр. весов. Ф-ция регрессии $\sum (\lambda_i^- - \lambda_i^+) k(x_i, x) - w_0$

$w_0: (w, x_i) - w_0 = \begin{cases} y + \varepsilon, & \xi_i \in 2 \\ y - \varepsilon, & \xi_i \in 3 \end{cases}$

Классификация:

$\varphi: \mathbb{R}^n \to H$, H - вект. пр-во. сразу сведем задачу к лин. раздел. выборке
раздел. ф-ция $f(x) = (w, \varphi(x)) + b \quad w = \sum \lambda_i y_i \varphi(x_i)$, где λ_i зависит от y_i и $(\varphi(x_i), \varphi(x_j))$, рассм. $k(x, y) = (\varphi(x), \varphi(y))$, k неотр. опр. и симм.

В случае лин. разд. выборки: ищем $f(x)$ т.ч. $f(x_i) > 0 \; \forall x_i \in W_1$, $f(x_i) < 0$
$\forall x \in W_2$, $y_i = \begin{cases} 1, & x \in \omega_1 \\ -1, & x \in \omega_2 \end{cases}$, т.е. $y_i((w, x_i) + b) \ge 0$

Разд. гиперповерхность $(w, x) + b = 0$, по т-ме Куна-Таккера задача экв. $h(w, b, \lambda) =$
$= 0{,}5 (w, w) - \sum \lambda_i (y_i ((w, x_i) + b) - 1) \to \min\limits_{w, b} \max\limits_{\lambda}$

$\lambda_i \geq 0$, $\lambda_i(y_i(\langle w, x_i\rangle + b) - 1) = 0$, откуда $\lambda_i = 0$ или (Стр. 6)
$y_i(\langle w, x_i\rangle + b) - 1 = 0$, из усл. сущ. седловой точки:

$$\begin{cases} \frac{\partial L}{\partial w_s} = w_s - \sum \lambda_i y_i x_{is} = 0 \\ \frac{\partial L}{\partial b} = -\sum \lambda_i y_i = 0 \end{cases} \Rightarrow w = \sum \lambda_i y_i x_i \text{ и } \sum \lambda_i y_i = 0$$

$L(w, b, \lambda) = \sum \lambda_i - 0{,}5 \sum \lambda_i \lambda_j y_i y_j (x_i, x_j) = \sum \lambda_i - 0{,}5 \|\sum \lambda_i y_i x_i\|^2$ т.е.
сводим и получаем итоги $\Phi(\lambda) = \sum \lambda_i - 0{,}5 \|\sum \lambda_i y_i x_i\|^2$

Kernel trick
$w = \sum \lambda_i y_i \varphi(x_i)$ $\arg\max f(\lambda_1 \dots \lambda_n) = \sum \lambda_i - \frac{1}{2} \sum_i \sum_j y_i \lambda_i k(x_i, x_j) y_j \lambda_j$
$\sum \lambda_i y_i = 0$, $0 \leq \lambda_i \leq \frac{1}{2n\lambda}$ $\forall i$, $\varphi(x_i)$ лежит на гран. в преобр. пр-ве
ищем $b = w^T \varphi(x_i) - y_i = \sum y_j \lambda_j k(x_j, x_i) - y_i$; $z \mapsto \mathrm{sgn}(w^T \varphi(z) - b)$

Билет 8

Метод k-ближайших соседей. Оценка оптимальности классификации методом kNN (нер-во Ковера-Харта)

Рассм. train $X^m = \{(x_i, y_i)\}_{i=1}^m$, метрика $\rho(x, x')$ для прогн. объекта, располагаем объекты x_i в порядке возр. до u: $\rho(u, x_{1;u}) \leq \dots \leq \rho(u, x_{m;u})$, $x_{i;u}$ – i-й сосед и тогда алгоритм будет задаваться как $a(u) = \arg\max_{y \in Y} \sum [y_{i;u} = y] w(i,u)$, где $w(i,u)$ оценивает степень важности i-го соседа для класс. u. Для классического kNN $w(i,u) = [i \leq k]$, при плавном $w(i,u)$ нач-во м.б. много, тогда применяют ядро. Пример: $w(i,u) = K\left(\frac{\rho(u, x_{i;u})}{h}\right)$ – парзеновское окно
Н-во Ковера-Харта: верхняя граница частоты ошибок $R^* \leq R_{kNN} \leq R^*\left(2 - \frac{MR^*}{M-1}\right)$, R^* – частота ошибок Bk, R_{kNN} – асимпт. частота, M – кол-во классов

Билет 9

Задача классификации. Основные метрики (confusion matrix, ROC AUC) класс. Оценка обобщ. способности модели.

~~Задача~~ Рассм. зад. бин. классификации $X \to Y$, $X = \mathbb{R}^n$, $Y = \{-1, +1\}$, $X^m = (x_i, y_i)_{i=1}^m$, класс. $a = \mathrm{sgn}(\langle w, x\rangle - w_0)$, ищем параметры w, мин. риск. $R(X^m, w, w_0) = \sum [a \neq y_i] = \sum [(\langle w, x_i\rangle - w_0) y_i < 0]$
$\text{accuracy} = \frac{1}{l} \sum_{i=1}^{l} [a(x_i) = y_i] = \frac{TP + TN}{FP + FN + TP + TN}$

Матр. ошибок:

	pred +	pred −
+	TP	FN
−	FP	TN

При большом w_0 много x_i т.ч. $a(x_i) = -1$, добавл. d_y – штрафы, тогда $\text{loss} = d_{y_i}[a(x_i) \neq y_i]$
ROC-AUC по x: $TPR = \frac{\sum [y_i = -1][a(x_i) = +1]}{\sum [y_i = -1]}$, по y: $TPR = \frac{\sum [y_i = +1][a(x_i) = +1]}{\sum [y_i = +1]}$. Оценки обобщ. способности: $X^l = \{(x_i, y_i)\}_{i=1}^l$, $y_i = y^*(x_i)$, $A_t = \{a: X \to Y\}$, $\mu_t: (X \times Y)^l \xrightarrow[\text{обуч.}]{\text{метод}} A_t$
$L(a, x)$ – loss, $Q(a, X^l) = \frac{1}{l} \sum L(a, x_i)$ – ф-ционал кач-ва
Самый common: hold-out $X^L = X^l_n \sqcup X^k_n$, $n = 1 \dots N$
$\text{cross-val}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q_\mu(X^l_n, X^k_n)$