Topics:

- Our prediction schema
- K-fold cross-validation
- Forward and Backward phase
- Chain Rule (on board)
- Derivation how compute partial derivatives for quadratic Loss
- Source code with which you figure out "What it do?"

Our Prediction Schema

$$\tilde{F}(x; a, b) = \sum_{m=0}^{M} b_m \cdot S_m\left(\sum_{j=0}^{n} a_{jm} x_j\right)$$

1. $S_0(z) = 1$ always.
   And => $S_0' = 0 = S_0(1 - S_0)$
1. $S(z)' = S(z)(1 - S(z))$

K fold cross-validation

- One way to mitigate problems with simple cross-validation is use K-fold cross-validation.
- This technic allow to have a data manipulating schema when number of observation is not very big or even limited and we can not make train set and test set as big as we want.
- **Algorithm:**

1. Split available data into K disjoint folds(or groups, or subsets): $D_1, D_2, \ldots, D_k$ which are in the union gives original available data. For example typical case K = 10

2. For $= \overline{1, K}$:
Train each Predictor (or Model) on Train $(\bigcup_i D_i)/D_j$ and evaluate empirical loss on Test D_j.

3. For each predictor average empirical loss on all test set D_j. It is an estimation of generalization error.

Here cross-validation estimate the performance of the actual predicting.
4.Select a model with lowest generalization error. One possible way of doing things if there are several models with small generalization error - select “simplest” one

What is forward propagation?

- Take out model and subsitute x and evaluate $\overbrace{F}(x; a, b)$

- If need store some intermediate results then do it

- Motivation video with “Do it”

Loss function for regression for NN

- $L = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \widetilde{F}(x_i)\right)^2$ and even more $\frac{1}{N}$ multiplier can be omitted.
- And also people who create neural networks approximations consider $L = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \widetilde{F}(x_i)\right)^2$.
- For differentiable function negative gradient show a direction to fastest local descend of function value if choose direction from unit norm sphere where norm is Euclidian norm.

Chain Rule

- In board discuss Chain Rule
- Gradients
- Linearity of apply derivative
- Difference between Gradient and Jacobian

Derivative to compute partial derivatives

For single layer network and for single observation we have

$$\mathrm{r}_i = y_i - \overset{\frown}{F}(x_i)$$

$$L = \frac{1}{2}\left(y_i - \overset{\frown}{F}(x_i)\right)^2 = \frac{1}{2}r^2$$

$$\frac{dL}{dr} = \frac{2}{2}r = r = (y_i - \overset{\frown}{F}(x_i))$$

$$\overset{\frown}{F}(x; {\color{red}a, b}) = \sum_{m=0}^{M} b_m \cdot S_m\left(\sum_{j=0}^{n} a_{jm}x_j\right)$$

$$\frac{\partial L}{\partial b_m} = \frac{dL}{dr} \cdot \frac{\partial r}{\partial b_m} = \left(y_i - \overset{\frown}{F}(x_i)\right) \cdot \frac{d\left(-\overset{\frown}{F}(x_i)\right)}{db_m} = \left(\overset{\frown}{F}(x_i) - y_i\right) \cdot S_m\left(\sum_{j=0}^{n} a_{jm}x_j\right)$$

$$\frac{\partial L}{\partial b_m} = \text{"}error\text{"} \cdot \text{"}output\ of\ the\ m'th\ internal\ node\text{"}$$

Derivative to compute partial derivatives

For single layer network and for single observation we have

$$\mathrm{r}_i = y_i - \widetilde{F}(x_i)$$

$$L = \frac{1}{2}\left(y_i - \widetilde{F}(x_i)\right)^2 = \frac{1}{2}r^2$$

$$\frac{dL}{dr} = \frac{2}{2}r = r = (y_i - \widetilde{F}(x_i))$$

$$\widetilde{F}(x; a, b) = \sum_{m=0}^{M} b_m \cdot S_m\left(\sum_{j\prime=0}^{n} a_{j\prime m} x_{j\prime}\right)$$

$$\frac{\partial L}{\partial a_{jm}} = \frac{dL}{dr}\frac{\partial r}{\partial a_{jm}} = (y_i - \widetilde{F}(x_i))\frac{\partial\left(y_i - \widetilde{F}(x_i)\right)}{\partial a_{jm}} = \left(\widetilde{F}(x_i) - y_i\right) b_m \cdot \frac{\partial\left(S_m\left(\sum_{j'=0}^{n} a_{j'm} x_{j'}\right)\right)}{\partial a_{jm}}$$

$$\frac{\partial L}{\partial a_{jm}} = (\widetilde{F}(x_i) - y_i) b_m \cdot S\left(\sum_{j\prime=0}^{n} a_{j\prime m} x_{j\prime}\right)\left(1 - S\left(\sum_{j\prime=0}^{n} a_{j\prime m} x_{j\prime}\right)\right) \cdot x_j$$

Practice

- L4_single_nn_compute_gradient_of_loss.py
- Read it and fill comments (In Russian on English)