CS F425: Deep Learning

11 Neural Weight Initialization

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 14, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

http://ktiwari.in/dl

One Hot Encoding

How should we encode the output?

- Assume there are five class cat, dog, bus, plant, apple

Class	Code-1	Code-2
cat	001	10000
dog	010	01000
bus	011	00100
plant	100	00010
apple	101	00001

Consider Code-1:

- Average of cat and bus?
- Determine similarity in $sim(cat, dog)$ and $sim(cat, bus)$
- Does the loss for plant and apple is same?

Issues leading to Code-2 (One Hot Encoding):

- Ordinal relationship should not be present
- Similarity between classes to be avoided (make them orthogonal)
- Loss should not favour (be biased) to any class

Deep Learning Approach

DL is constructing network (neural obviously)
of parameterized functional modules
and training then from examples
using gradient based optimization

- Yun LeCun

There are many interesting questions

- How many Layers?
- How many units in each layer?
- What should be the learning rate?
- How to initialize the parameters?
- What is right activation function? etc.

Hyperparameter Tuning

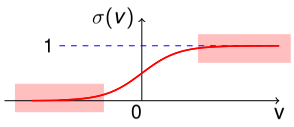
It is an iterative process.

- Where is the nob?
  Underfitting/Overfitting
- Bad everywhere (high bias), vs on test data (high variance)
- Go towards more complex model, or increase data/regularization (reduce complexity)

Exploding and Vanishing Gradient

- If weights are too small, variance of the input signal starts diminishing as it passes through layers.
- If weights are too large, variance get huge amplification.

$\sigma(v) = \frac{1}{1+e^{-v}}$



$\sigma(+ve) > 0.5$
$\sigma(-ve) < 0.5$

We want variance of input and output should not differ much (atleast initially)

(0.90→0.81→0.73→0.66→0.59→0.53→0.48→0.43→0.39) (1.50→2.25→3.38→5.06→7.59→11.39→17.09→25.63)

Xavier Weight Initialization

- Random initial W, likely tend to exploding and vanishing gradient
- Initialization with zero weights makes things linear (as all neurons become symmetric).
- Standard is to make weights centered around 0 with small variance[1] k/S_{l-1} where previous layer has S_{l-1} neurons[2]
- Weights are sampled on normal distribution with varience k/S_{l-1}
  $k = 2$ for RelU and $k = 1$ for sigmoid/tanh
- This is called xavier initialization
- Another possibility is to use variance as $k/(S_{l-1} + S_l)$ or $k/(S_{l-1}S_l)$

Apply **gradient clipping** if needed.

[1] variance is average of the squared differences from the mean
[2] Standard daviation becomes $\sqrt{k/S_{l-1}}$

Xavier Initialization

Assume data is i.i.d. from normal distribution with $\mu = 0, \sigma = 1$

- As $var[w_i x_i] = E[x_i]^2 var[w_i] + E[w_i]^2 var[x_i] + var[x_i]var[w_i]$ since $E[x_i] = \mu = 0$ therefore $var[w_i x_i] = var[x_i]var[w_i]$
- Consider a neuron $y = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b$

$$\begin{aligned} var[y] &= var[w_1 x_1 + w_2 x_2 + ... + w_n x_n + b] \\ &= var[w_1]var[x_1] + var[w_2]var[x_2] + ... + var[w_n]var[x_n] + var[b] \\ &= var[w_i]var[x_i] + var[w_i]var[x_i] + ... + var[w_i]var[x_i] + 0 \\ &= S_{l-1}.var[w_i]var[x_i] \qquad n = S_{l-1} \end{aligned}$$

To have same varince in input and output $S_{l-1}.var[w_i] = 1$ or

$$var[w_i] = 1/S_{l-1}$$

Thank You!

Thank you very much for your attention!