CS F425: Deep Learning

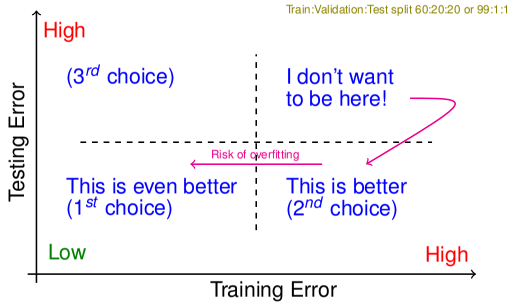
12 Regularization For Neural NW

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

Feb 16, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

http://ktiwari.in/dl

Which side do you want to be?



Low training error comes with a risk of overfitting (high variance)

Recall Regularization for Logistic Regression

- Optimization minimizes the loss $min_w J(w)$ by adjusting w where

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} Loss(\hat{y}^{(i)}, y^{(i)})$$

- Regularization penalizes the large values of w by

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} Loss(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}||w||_2^2$$

where

$$||w||_2^2 = \sum_j w_j^2 = w^T w$$

λ being regularization parameter

Regularization for Neural Network

- As there could be L layers, each having own set of parameters, so

$$J(w^{[1]}, w^{[2]}, ..., w^{[L]}) = \frac{1}{m}\sum_{i=1}^{m} Loss(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L} ||w^{[l]}||_F^2$$

where frobenius norm is

$$||w^{[l]}||_F^2 = \sum_{i=1}^{n^{[l-1]}}\sum_{j=1}^{n^{[l]}} w_{ij}^2$$

- How you updated the parameter earlier? get $dw^{[l]} = (from\ backpropogation)$ that is $\frac{\partial J}{\partial w^{[l]}}$ then

$$w^{[l]} = w^{[l]} - \alpha.dw^{[l]}$$

Weight Decay: Regularization for NN (contd..)

- With regularization term $dw^{[l]} = (from\ backpropogation) + \frac{\lambda}{m}w^{[l]}$
- Therefore the update is modified to

$$w^{[l]} = w^{[l]} - \alpha.((from\ backpropogation) + \frac{\lambda}{m}w^{[l]})$$

Which is

$$w^{[l]} = \left(1 - \frac{\alpha\lambda}{m}\right)w^{[l]} - \alpha.(from\ backpropogation)$$

- Due to the $(1 - \frac{\alpha\lambda}{m})$ factor, this update method is also called weight decay

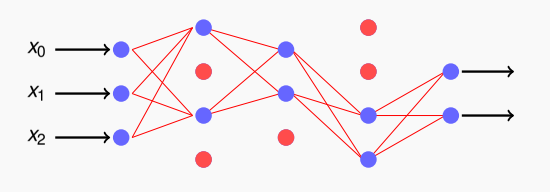
Our objective here is to penalize the weight matrices being too large.

Penalize the weight matrices from being too large

- With low weight, the connection get weakened so network has effectively less connections and become simpler.
- Another intuition is that, when weights are lower, the output of the units is also lower. Assume activation function be `tanh` small input values tends to produce linear output.
- So overall n/w tends to becomes linear (more biased).

Dropout Regularization

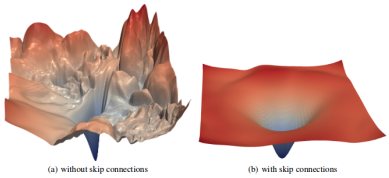
Shutdown some units randomly



- Drop probabilities for different layers may be different
- Now networks cannot rely on any specific connection. So have to give importance to others neuron also
- Issue is that cost function in now not well defined

Regularization

- Dropout[1]



(a) without skip connections (b) with skip connections

- Early stopping

[1] Visualizing the Loss Landscape of Neural Nets, Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein, 2018

Other Regularization Methods

- Increase the data by data augmentation
- Flip, rotate, scale, translate, deform ....

- Normalize training examples $\rightarrow$ to speed up your training

Thank You!

Thank you very much for your attention!