L07: 05.05.2022

Layer L-1

$$\frac{\partial L}{\partial w_{L-1,ij}} = \frac{\partial L}{\partial \underline{a}_L} \cdot \frac{\partial \underline{a}_L}{\partial \underline{a}_{L-1}} \cdot \frac{\partial \underline{a}_{L-1}}{\partial w_{L-1,ij}}$$

$$= \underbrace{\underline{\underline{J}}_L(\underline{a}_L) \cdot \underline{\underline{J}}_{\underline{a}_L}(\underline{a}_{L-1})}_{\underline{\underline{J}}_L(\underline{a}_{L-1})} \cdot \underline{\underline{J}}_{\underline{a}_{L-1}}(w_{L-1,ij})$$

$1 \times M_{L-1}$ $\qquad$ $M_{L-1} \times 1$

$\vdots$

Layer 1:

$$\frac{\partial L}{\partial w_{1,ij}} = \underbrace{\underline{\underline{J}}_L(\underline{a}_L)\, \underline{\underline{J}}_{\underline{a}_{L-1}}(\underline{a}_{L-1}) \cdots \underline{\underline{J}}_{\underline{a}_2}(\underline{a}_1)}_{\underline{\underline{J}}_L(\underline{a}_1)\ 1 \times M_1} \underline{\underline{J}}_{\underline{a}_1}(w_{1,ij}) \qquad M_1 \times 1$$

Similarly,

$$\frac{\partial L}{\partial b_{1,i}} = \underline{\underline{J}}_L(\underline{a}_1) \cdots \underline{\underline{J}}_{\underline{a}_1}(b_{1,i})$$

This is backpropagation

Slide 4-23 in each update step

4.28.1

more about optimizer : Ch 5

" " model : Ch 6-9

Batch gradient descent : Calculate $\nabla L(\underline{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \nabla l(\underline{x}(n), y(n); \underline{\theta})$

for the whole training set D_{train}

Difficulty :

N maybe too large

eg MNIST: 60 k

ILSVRC: 1.2 million

⇒ not enough GPU memory for D_{train}, $\underline{\theta}$, $\{\underline{a}_l, \underline{x}_l\}$,

gradients, Jacobi matrixes

Solution :

Minibatch and Stochastic gradient descent (SGD)

Calculate ∇L and update for $\underline{\theta}$ for one minibatch,

minibatch - a block of B samples from D_{train}, $B \ll N$:

$$\underline{\theta}^{t+1} = \underline{\theta}^t - \gamma^t \nabla L(t, \underline{\theta} | \underline{\theta} = \underline{\theta}^t)$$

$$L(t, \underline{\theta}) = \frac{1}{B} \cdot \sum_{n=t'B+1}^{(t'+1)B} l(\underline{x}(n), \underline{y}(n); \underline{\theta})$$

$B \in \mathbb{N}$: mini batch size, small enough for GPU memory

N/B : no of minibatches in D_{train}

$t = 0, 1 \cdots$: iteration index - update step

$t' = 0, 1 \cdots N/B - 1$: minibatch index

Slide 4-29

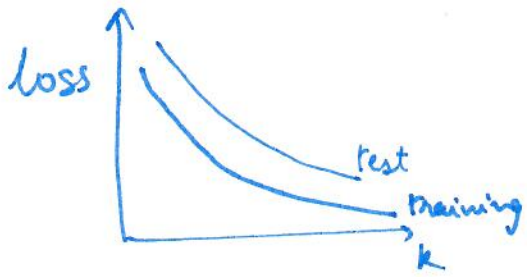
$\nabla L(t; \underline{\theta})$ more noisy than $\nabla L(\underline{\theta})$, hence stochastic gradient descent.

Evaluation During Training

$\hat{\underline{\theta}}_k$ be the predicted o/p after each epoch $k = 1, 2, \ldots$

Technical metrics :

- training loss : $L(\hat{\underline{\theta}}_k)$ calculated from training set D_{train} using fwd pass
- test loss : $L(\hat{\underline{\theta}}_k)$ calculated from test set D_{test}



More objective metrics for classification

- training error rate : Error rate of DNN $(\hat{\underline{\theta}}_k)$ from D_{train}
- test error rate : Error rate of DNN $(\hat{\underline{\theta}}_k)$ from D_{test}

or

accuracy = 1 - error rate

4.8 Implementation of DNNs in Python

slide 4.30

Large gap b/w training and testing → overfitting