ConvNets

Sunday, 1 April 2018 09:51

Simple → complex. with the layer goes on.

$G_1(32\times32\times3)$ $\xrightarrow{\text{conv ReLU}}$ $G_1'(28\times28\times6)$ $\xrightarrow{\text{conv ReLU}}$ $G_1''(24\times24\times10)$

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

element-wise multiplication and sum of a filter and the signal.

Output size: $(N-F)/\text{stride} + 1$

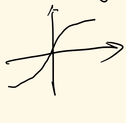
Pooling Layer. => down sampling.

max pooling: find max in a specific region.

Activation Functions:

$\sigma(x) = 1/(1+e^{-x})$

problem: kill the gradient.

tan $h(x)$:  0 centered but still kill gradient.

dead ReLU: Some will never be updated when $x<0$.

let b is 0.01 such small number may help.

Leaky ReLU: $f(x) = \max(0.01x, x)$

Parametrix: $\max(\alpha x, x)$

ELU: $f(x) = \begin{cases} x & \text{if } x>0 \\ \alpha(e^x-1) & \text{if } x\le 0 \end{cases}$ closer to 0 mean.

Maxout:

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

Don't use sigmoid

Not common to normalize variance, to do PCA or whitening.

Xavier initialization

Batch Normalization.

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$$

Input: values of x over a mini-batch $B=\{x_{1\dots m}\}$
parameters to be learned: γ, β.

Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_B)^2$$

$$\hat{x}_i \leftarrow \frac{x_i-\mu_B}{\sqrt{\sigma_B+\epsilon}}$$

// for hype parameter: random search usually better than grid search