S20180010017 Set-A

ASHUTOSH CHAUHAN

DEEP-LEARNING

Q.1.) Given objective function $\rightarrow -\sigma(y_i w^T x_i)$ | negative log likelihood.

a) Gradient update $\rightarrow$ $w \leftarrow w - \eta \Delta \hat{L}(w)$

$$\Delta L \rightarrow \frac{d \log(f(x))}{dx} \rightarrow \frac{f'(x)}{f(x)}$$

$$\Delta L \rightarrow \frac{\sigma'(y_i w^T x_i)}{\sigma(y_i w^T x_i)}$$

$$\Delta L \rightarrow \frac{\cancel{\sigma(y_i w^T x_i)}\,\sigma(-y_i w^T x_i)\cdot y_i x_i}{\cancel{\sigma(y_i w^T x_i)}} \quad \Big|\ \text{w.r.t } w$$

$$\boxed{w \leftarrow w + \eta(\sigma(y_i w^T x_i))}$$

b.) In Non-Linear SVM we use the kernal function to do the computation equivalent to doing the transformation of Input space to higher dimensions and doing the mathematical operation leading to significant drop in computation.

Kernal trick

Consider a transformation of input into a higher dimension ϕ. Then we can use a kernal function $K(x, x') = \phi(x)^T \phi(x')$ provided $K(x, x')$ is positive definate and symmetric.

This allows us to have a non-linear hyperplane in the original input space and do not require transformation of input explicitly.

Ashutosh

S2018 001 0017
Ashutosh Chauhan

Set- A

Q.2.)

(a)

$L(x) \rightarrow \underline{2x_1 - 4x_2 + x_3} \rightarrow 1$ or 0.

Pattern	P_1	P_2	P_3	P_4
x_1	1	0	1	1
x_2	0	1	0	1
x_3	0	1	1	1
Output	1	0	1	0

$\rightarrow$

Q.2

(b)

(i) Input $\rightarrow$ $64 \times 64 \times 10$, $\rightarrow$ filter $n = 20$, 5×5, stride 5, pad 3

Size of output $\Rightarrow$ width $\Rightarrow \left(\frac{64 - 5 + 6}{5}\right) + 1 \rightarrow 14$

$\boxed{14 \times 14 \times 20}$

Number of parameter. $\rightarrow \underbrace{5 \times 5}_{\text{Spatial}} \times \underbrace{20}_{\text{n.f.}} \times \underbrace{10}_{\text{I.d.}} \rightarrow \underline{5000}$

(ii) Input $128 \times 128 \times 3$ $\rightarrow$ filter $n = 5$, 3×3 stride = 1, pad 1

output size, width $\rightarrow (128 - 3 + 2) + 1 \rightarrow \underline{128}$

$\boxed{128 \times 128 \times 5}$

no. of params $\rightarrow$ $5 \times \underline{3 \times 3} \times 3 \rightarrow \underline{135}$.

(iii) Input $30 \times 30 \times 3$ +
Batch normalization
Output size $\rightarrow$ Same $(30 \times 30 \times 3)$

no. of parameters (learnable) = 0 since its
Computed for each batch of inputs.

Ashutosh

52018001 0017

Set - A

Ashutosh Chauhan

Q.3

(a) Dropout allows us to ignore nuerons which might be learning features which are not required and increases the accuraty and confidence of the model.
Also it redirces the time required to test, etc since the no. of computations will be reduced since the model contains lesser number of nuerons now.

We use inverse dropout we use dropout at training time as well.

b.) Adam optimizer for gradient update.

$dw \leftarrow gradient(w)$

$m_1 \leftarrow \beta_1 * fm_1 + (1-\beta_1) * dw$

$m_2 \leftarrow \beta_2 * m_2 + (1-\beta_2) * dw * dw$

$w \leftarrow w - \alpha(m_1 / \sqrt{m_2 + 0.0000001})$

$w \rightarrow$ weight

$\beta_1, \beta_2 \rightarrow$ params

$m_1 \rightarrow$ first moment

$m_2 \rightarrow$ second moment.

$\alpha \rightarrow$ learning rate.

Ashutosh

Set - A

S2018 00 10017
Ashutosh Chauhan

Q.4.)

b.) using 3 3x3 kernal layers has the advantage to a single 7x7 kernal.

* using multiple kernals increases the non-linearity of than a single layer.
* requires less computation due to smaller size convolutions.
  i.e. (3x3 x3) < (7x7x1)
* covers the same spatial area as the 7x7 kernal.

a.) In Squeeze and excitation network we assign a priority value to each of the channels in the previous layer. i.e. scaling the channel wise wieghts.
it allows the network to learn the more important features and give less weight to lesser important ones.

Ashutosh