Deep Learning Mid Sem

Set-C

Sayam Kumar
S20180158

Q.1 Classifiers

(a) K-Nearest Neighbor classifier is a non-parametric algorithm for classification purposes. The idea is

i) Loop over all training examples and find K nearest neighbours using some distance (Euclidean/Manhattan)

ii) Pick the class label for data point as the one with majority label in K neighbours

* K is no of neighbours to select for predicting the label of test point.
* There is no training required in KNN
* Eg → If we are given weights and asked to classify as obese or not obese. Then for a test point, select (K=10) neighbours and select the major label in these 10 neighbours.

(b) SVM → Margin is the distance between support vectors and boundary hyperplane. Margin is meant to be maximized in SVM to pick the best classifier

$$l = \frac{\lambda}{2n}|W|^2 + \max(0, 1 - y_i w^T x_i)$$

finding $\frac{\partial}{\partial a}\max(0,a) = \begin{cases} 0 & \text{for } a < 0 \\ 1 & \text{for } a > 1 \end{cases} = \mathbb{I}[a > 0]$

$$\nabla_w l = \frac{\lambda w}{n} + \mathbb{I}[1 - y_i w^T x_i > 0](-y_i x_i)$$

Next Page

$$\nabla_w L = \frac{\lambda w}{n} - y_i x_i \, \mathbb{I}[y_i w^T x_i < 1]$$

update rule

~~$w \leftarrow w + \eta \nabla_w L$~~ $\quad w \leftarrow w - \eta \nabla_w L$

~~$\theta = w + \eta$~~ $\quad w = w - \eta \frac{\lambda w}{n} + y_i x_i \, \mathbb{I}[y_i w^T x_i < 1] \eta$

$$= \left(1 - \eta \frac{\lambda}{n}\right) w + \eta \, \mathbb{I}[y_i w^T x_i < 1] y_i x_i$$

In other words

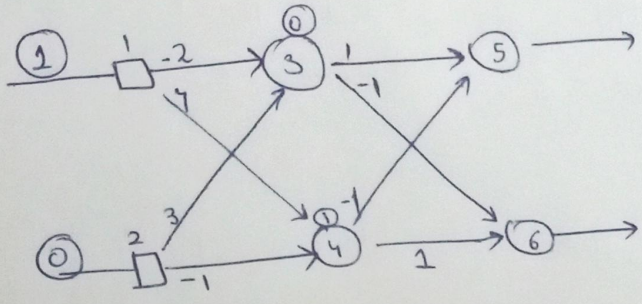
if $y_i w^T x_i < 1$

$$w = w + \eta \left(y_i x_i - \frac{\lambda}{n} w\right)$$

else

$$w = w - \eta \frac{\lambda}{n} w$$

Q-2 Neural Network



$In_3 = -2 + 0 = 2 \Rightarrow out3 = 0$

$In_4 = 4 + 0 = 4 \Rightarrow out4 = 1$

$In5 = 0 + -1 = -1 \Rightarrow out5 = 0$

$In6 = 0 + 1 = 1 \Rightarrow out6 = 1$

$y_5 = 0$

$y_6 = 1$

$y_5 = 0$

$y_6 = 1$

Q-2 (b) Input Volume

$30 \times 30 \times 256$

Max pool 2×2 Stride 2 Pad 0

$$\text{Output size} = \frac{30-2}{2} + 1 = 15 \times 15$$

Output size $= 15 \times 15$, No of params $= 0$

ii) Input $128 \times 128 \times 3$ Batch Norm Layer

No shape change

Output shape $128 \times 128 \times 3$

No of parameters $= \gamma, \beta$ (learnable) $= 2$

(assuming full batch update)

iii) Input $16 \times 16 \times 16$

filters 5×5 (8 filters) 2 Stride 2 Pad

$$W = \frac{16-5+2(2)}{2} + 1 = \frac{11+4}{2} + 1 = \frac{15}{2} + 1 = [8.5] = 8$$

~~Cannot apply this~~

Output size $= 8 \times 8 \times 8$

Params $= (5 \times 5 \times 16) \times 8$ weights + 8 bias ~~filter~~ $= 3200$ weights + 8 bias

---

Q-3 (a) Activation fuctions are used to introduce non linearities in ~~d~~ training and extracting features from data. It helps in activating and proper gradient flow during training

Next Page

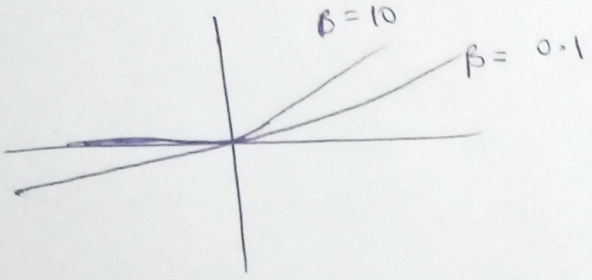
Swish activation function is

$f(x) = x \cdot \text{sigmoid}(\beta x)$

where β is learnable parameter / hyperparameter

ReLU is a special case of swish (ReLU is with high values of Beta)

graph



(iv) Adagrad Optimizer → Adagrad is for smoothing out the descent when reaching the optima (minima) value

while True:

$dx \leftarrow \text{gradient}(x)$

$\text{grad\_sq} \leftarrow \text{grad\_sq} + (dx) * (dx)$

$$x \leftarrow x - \eta \times \frac{dx}{\sqrt{\text{grad\_sq} + 1e^{-7}}}$$

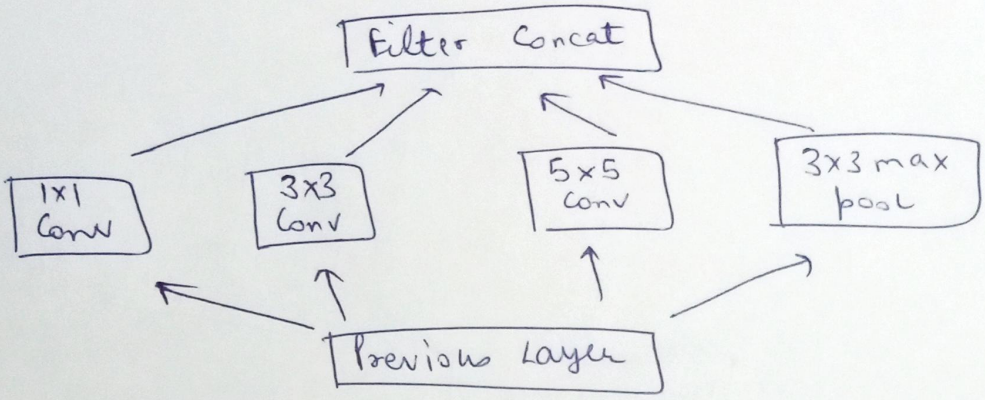
Drawback → Overtime the grad-sq goes on increasing, thereby decreasing the learning rate before reaching minima. This is also called effective learning rate diminishing problem

RMS prop resolves this problem by a (decay_rate) parameter that accounts for weighted average of ∇f^2 and helps in mini batch training.

Sayam Kumar
S20180010158
Page 5

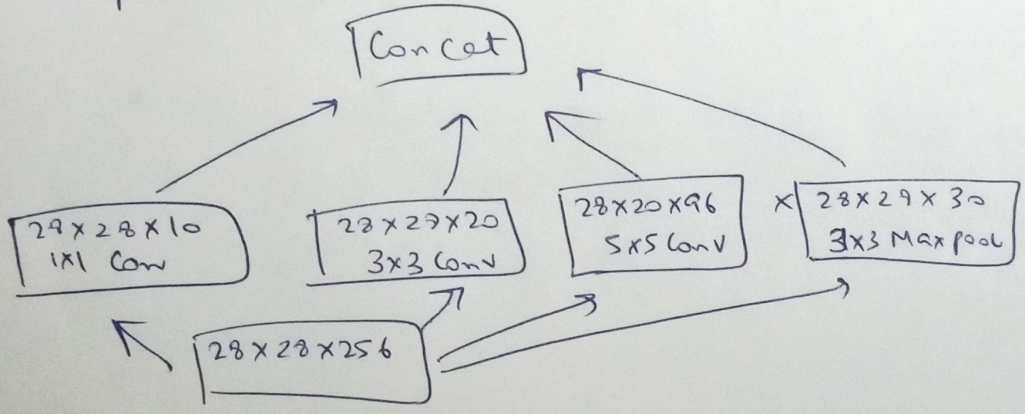
Q-4 Inception Module

Inception is about using multiple filters with different sizes stacked together. This a deeper network but a lot computational efficiency.



It has a problem of computation with too many multiplication operations.

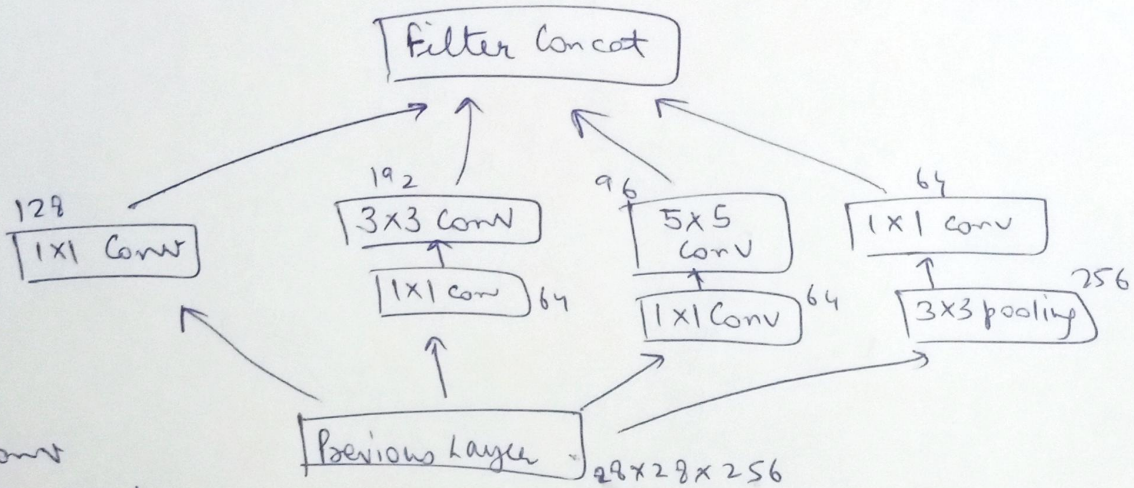
Taking example



Total about 600M operations

Expensive

(b) Bottleneck is used with 1×1 conv layer to reduce the computational complexity of Inception module



1×1 conv filters used as bottleneck.

Operations are dramatically reduced from 850M ops to 350M operations.

1×1 conv filters helps in better weighted average for gradient flow and no of operations.