Neural Computation

General Loss Function = (predicted output - original output)^2.

underfitting = Training ↓ Testing ↓ Performance.

Overfitting = Training ↑ Testing ↓ Performance

Optimum = Training 基本↑ Testing ↑ Performance.

Data (100%)	
Training (80%)	Testing (20%)
Train (80%) / Valid (20%)	Testing (20%)

Linear Model

$$C(\omega) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \omega^T x^{(i)})^2 \qquad C'(\omega) = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)})^2\omega - \frac{1}{n}\sum_{i=1}^{n} y^{(i)}x^{(i)}$$

$$= \frac{1}{2n}(\omega^T X^T X\omega - 2\omega^T X^T y + y^T y)$$

ω^* is the minimum point.

First order optimal condition: $\nabla C(\omega^*) = 0$

One Dimension case: ~~$C'(\omega)$~~ $\omega^* = \frac{\sum_{i=1}^{n} y^{(i)}x^{(i)}}{\sum_{i=1}^{n}(x^{(i)})^2}$

$(X\omega)^T = \omega^T X^T$ Closed form: $\omega^* = \arg\min_{\omega} C(\omega) \Rightarrow \omega^* = (X^TX)^{-1}X^Ty$

Polynomial regression

$$f(x) = \omega^T\phi(x) = \omega_0 + \omega_1 x + \omega_2 x^2 + \omega_3 x^3.$$

$$\omega^* = (X^TX)^{-1}X^Ty$$

$$\hat{y} = X\omega^*$$

Regularized Regression.

$$C(\omega) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \omega^T x^{(i)})^2 + \frac{\lambda}{2}\|\omega\|_2^2 \qquad \omega^* = \left(\frac{1}{n}(X^TX) + \lambda I\right)^{-1}\left(\frac{1}{n}X^Ty\right)$$

Gradient Descent

$$w^{t+1} = w^t - \eta \nabla C(w^t)$$

GD For Regression

$$C(w) = \frac{1}{2n} (w^T x^T x w - 2 w^T x^T y + y^T y)$$

$$\nabla C(w) = \frac{1}{n} (x^T x w - x^T y)$$

$$w^{(t+1)} = w^{(t)} - \frac{\eta}{n} (x^T x w - x^T y)$$

Stochastic Gradient Descent

$$C(w) = \frac{1}{n} \sum_{i=1}^{n} C_i(w)$$

Learning rate $\eta_t = c/\sqrt{t}$, c is a parameter needed to tune.

Minibatch SGD

$$w^{(t+1)} = w^{(t)} - \frac{\eta_t}{b} \sum_{i \in B_t} \nabla C_i(w^{(t)})$$

Linear Model for Classification

$$\hat{y} = \text{sgn}(w^T x) = \begin{cases} 1 & w^T x > 0 \\ -1 & \text{otherwise} \end{cases}.$$

Margin $= y w^T x$

$$C(w) = \frac{1}{2n} \sum_{i=1}^{n} (\max\{0, 1 - y^{(i)} w^T x^{(i)}\})^2$$

$$\nabla Q(w) = \begin{cases} 0 & \text{if } y^{(i)} w^T x^{(i)} \geq 1 \\ (w^T x^{(i)} - y^{(i)}) x^{(i)} & \text{otherwise.} \end{cases}$$

SGD for Linear Model

$$C_i(w) = \begin{cases} 0 & \text{if } y^{(i)} w^T x^{(i)} \geq 1 \\ \frac{1}{2}(1 - y^{(i)} w^T x^{(i)})^2 & \text{otherwise.} \end{cases}$$

$$w^{t+1} = w^t - \eta \nabla C_i(w^t) = \begin{cases} w^{(t)} & \text{if } y^{(i)} (w^{(t)})^T x^{(i)} \geq 1 \\ w^{(t)} - \eta y^{(i)} (y^{(i)} w^T x^{(i)} - 1) x^{(i)} & \text{otherwise.} \end{cases}$$

Perceptron

if $y w^T x \leq 0$ then $w = w + yx$.

$$y = \sigma\left(\sum w_i x_i + b\right)$$

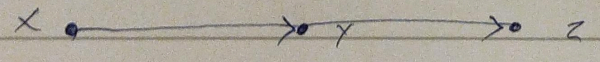
$$C(w) = \frac{1}{2n} \sum_{i=1}^{n} \left(\sigma(w^T x^{(i)} + b) - y^{(i)}\right)^2$$

Activation Functions

Sigmoid $\sigma(x) = \frac{1}{1 + e^{-x}}$ $\quad (-\infty, \infty) \to (0, 1)$

Tanh $(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ $\quad (-\infty, \infty) \to (-1, 1)$

ReLU(x) $= \max(0, x)$ $\quad (-\infty, \infty) \to (0, \infty)$
(Not Differentiable)



$\frac{dz}{dx} = \frac{dz}{dy} \times \frac{dy}{dx}$ → local gradient.

↓

Back-propogation gradient. $\left(\delta_j^l = \frac{\partial C}{\partial z_j^l}\right)$

Backpropogation

$$\frac{\partial C}{\partial w^l} = \delta^l (a^{l-1})^T \qquad \frac{\partial C}{\partial b^l} = \delta^l.$$

Backpropagated Gradient for output Layer.

$$\delta_j^L = \sigma'(z_j^L)(a_j^L - y_j)$$

For Hidden Layer.

$$\delta_j^L = \sigma'(z_j^l) \sum_k \delta_k^{l+1} \omega_{kj}^{l+1}$$

GD with momentum

$$v^{(t+1)} = \alpha . v^{(t)} - \eta \nabla C(w^{(t)}) \qquad \alpha \in [0, 1)$$

$$w^{(t+1)} = w^{(t)} + v^{(t+1)}$$

Nesterov Momentum (Nesterov Accelerated Gradient).

$$w^{(ahead)} = w^{(t)} + \alpha v^{(t)}$$

$$v^{(t+1)} = \alpha v^{(t)} - \eta \nabla C(w^{(ahead)})$$

$$w^{(t+1)} = w^{(t)} + v^{(t+1)}$$

SGD with Momentum

$$g(t) = \nabla C_i(w_t^{(t)})$$

$$v^{(t+1)} = \alpha v^{(t)} - \eta g(t)$$

$$w^{(t+1)} = w^{(t)} + v^{(t+1)}$$

SGD with Nesterov Momentum

$$w^{(ahead)} = w^{(t)} + \alpha v^{(t)}$$

$$g^{(ahead)} = \nabla C_i(w^{(ahead)})$$

$$v^{(t+1)} = \alpha v^{(t)} - \eta g^{(ahead)}$$

$$w^{(t+1)} = w^{(t)} + v^{(t+1)}$$

AdaGrad

$$g^{(t)} = \nabla C_i(\omega^{(t)})$$

accumulated gradient norm square $r^{(t+1)} = r^{(t)} + g^{(t)} \odot g^{(t)}$

$$\omega^{(t+1)} = \omega^{(t)} - \frac{\eta}{\delta + \sqrt{r^{(t+1)}}} \odot g^{(t)}$$

RMSProp

$$r^{(t+1)} = \beta \cdot r^{(t)} + (1-\beta) g^{(t)} \odot g^{(t)}$$

Adam

$$s^{(t+1)} = \beta_1 \cdot s^{(t)} + (1-\beta_1) g^{(t)}$$

$$r^{(t+1)} = \beta_2 r^{(t)} + (1-\beta_2) g^{(t)} \odot g^{(t)}$$

$$\hat{s}^{(t+1)} = \frac{s^{(t+1)}}{1-\beta_1^{t+1}} \qquad \hat{r}^{(t+1)} = \frac{r^{(t+1)}}{1-\beta_2^{(t+1)}}$$

$$\omega^{(t+1)} = \omega^{(t)} - \frac{\eta}{\delta + \sqrt{\hat{r}^{(t+1)}}} \odot \hat{s}^{(t+1)}$$

CNN Formula

$$\text{Output} = \frac{\text{Input} - \text{Kernel} + 2 \times \text{Padding}}{\text{Stride}} + 1$$

Depth = Number of filters.

Number of Parameters = Kernel x Kernel x Depth x no. of filters + no. of filters.

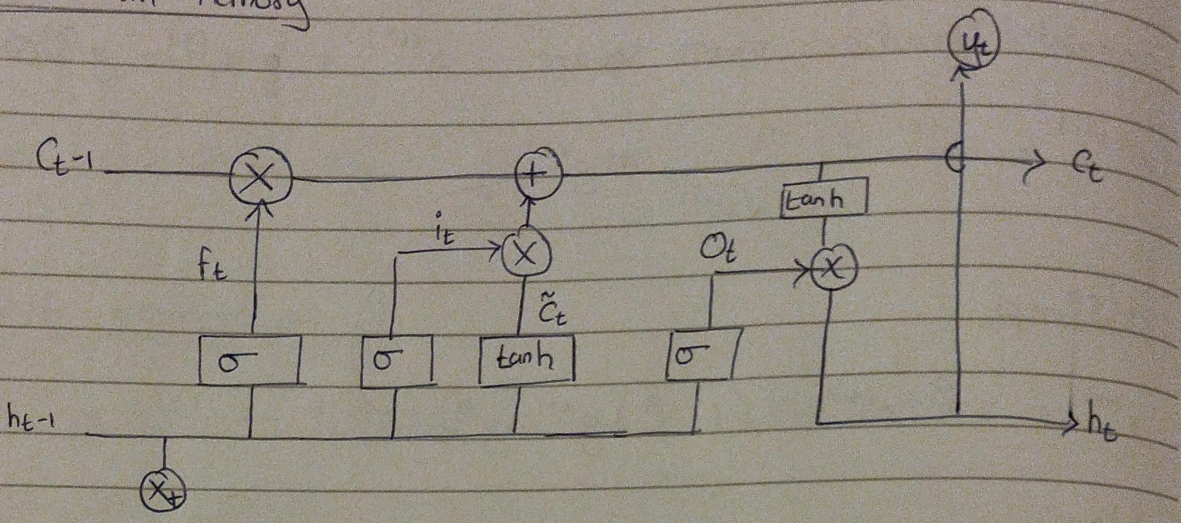
Upsampling

Output = (Input - 1) x Stride - 2 x Padding + (Kernel - 1) + 1 + output padding.

Depth = No. of filters.

Min - Max = $\frac{\text{value} - \text{min}}{\text{max} - \text{min}}$ z - score = $\frac{\text{value} - \text{mean}}{\text{std}}$.

Long Short Term Memory



① forget $f_t = \sigma(W_{fh}^T h_{t-1} + W_{fx}^T x_t)$

② Store $i_t = \sigma(W_{ih}^T h_{t-1} + W_{ix}^T x_t)$

$\tilde{C}_t = \tanh(W_{\tilde{C}h}^T h_{t-1} + W_{\tilde{C}x}^T x_t)$

③ Update $C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$

④ Output $O_t = \sigma(W_{oh}^T h_{t-1} + W_{ox}^T x_t)$

$h_t = O_t \times \tanh(C_t)$

$y_t = W_{hy}^T h_t$.

CNN Problem

Input = 11×11 Output = $5 \times 5 \times 10$

Kernel = 8×8 N = 10 Stride = 2 Padding = 1

Output = $4 \times 4 \times 5$ Kernel = $1 \times 1 \times 5$

$4 = \frac{5 - K + 2}{2} + 1$ H = 1 W = 1 O = 10 M = 5

Auto Encoders.

Loss function = $\frac{1}{d} \sum_{j=1}^{d} (x^{(j)} - g_\theta^{(j)}(f_\phi(x)))^2$.

Poobabilistic Encoder

$P_\phi(Z/x) = N(\mu_\phi(x), \sigma_\phi(x)^2)$.

VAE

$L_{VAE} = L_{rec} + \lambda L_{reg}$

↑ reconstruction loss ↖ Regulaizer.

$L_{rec} = \frac{1}{d} \sum_{j=1}^{d} (x^{(j)} - g_\theta(\tilde{z})^{(j)})^2$

Reparametesization Trick.

$\tilde{z} = N(\mu_\phi(x), \sigma_\phi(x)^2) = \mu\phi(x) + \sigma\phi(x) \odot \epsilon \quad \epsilon \sim N(0, I)$.

Practise

$C(w) = \frac{1}{2}(w^T x - y)^2$

$\nabla C(w) = (w^T x - y) x$ $\quad$ margin $= y w^T x$.

For Linear classification, $\nabla C(w) = (1 - \text{margin}) yx$

Paper

Q1. a) No. of neurons in Input Layer = 2.
Hidden = 2
Output = 1.

This is a fully connected NN.

No. of parameters = $[(2 \times 2) + 2] + [(2 \times 1) + 1] = 6 + 3 = 9$