01 मंगलवार Tuesday — फरवरी February

Regularisation

* help us solve the issue of overfitting.

Linear Regression :-

$$y = \underbrace{\theta_0}_{\text{bias}} + \theta_1 x_1 + \theta_2 x_2 + \cdots + \underbrace{\theta_n}_{\text{parameters/weights}} x_n = \theta^T x$$

$$= \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Total parameters :- n weights + 1 bias unit
↳ (n+1) parameters.

if n is very large there is a risk of overfitting
↓
- general solution to prevent overfitting - huge amt. of data containing all types of pattern.

1) L_1 Regularisation / Lasso - Least absolute shrinkage and selector operation

$$J_n(\theta) = J_0(\theta) + \underbrace{\alpha}_{\text{Regularisation factor } \in [0,1]} \underbrace{\sum_{i=1}^{m} |\theta_i|}_{\text{penalty term } L_1 \text{ term.}}$$

→ not containing bias

फरवरी February — बुधवार Wednesday 02

If during training at certain instants

$J_n(\theta) = 0 + \text{penalty} \Rightarrow$ error $\neq 0$ ↓ forced weight update

this noise will make it more generalised

While training

$J_n(\theta) = J_0(\theta) + L_1 \Rightarrow$ updated weights.

While testing we will not use the above
↳ $J_n(\theta) = J_0(\theta)$

फरवरी February — गुरुवार Thursday 03

Drawback :- 1) Latency due to extra calculations
2) $J_n(\theta)$ not differentiable everywhere.
3) α is an extra parameter to train

(ii) L_2 Regularisation / Ridge Regression.

$$J_n(\theta) = J_0(\theta) + \underbrace{\frac{\alpha}{2} \sum_{i=1}^{m} (\theta_i)^2}_{L_2 \text{ term}}$$

We are dividing by 2 as when we take $\frac{\partial}{\partial \theta} \theta^2 = 2\theta \times 1/2$

04 शुक्रवार Friday — फरवरी February

While training

$J_n(\theta) = J_0(\theta) + L_2$

while testing

$J_n(\theta) = J_0(\theta)$

Drawback

(1) Latency like before

(2) α is an extra parameter to tweak.

(iii) l_1-l_2 regularisation / elastic net

$$J_n(\theta) = J_0(\theta) + r\alpha \underbrace{\sum_{i=1}^{m} |\theta_i|}_{L_1} + (1-r)\frac{\alpha}{2} \underbrace{\sum_{i=1}^{m} \theta_i^2}_{L_2}$$

05 शनिवार Saturday — फरवरी February

r = controlling factor $\in [0,1]$

if $r \simeq 1$ l_1 will dominate → Lasso

$r \simeq 0$ l_2 will dominate → Ridge

(iv) Max Norm Regularisation

↳ we constraint the weights of incoming connections.

$\underbrace{\|w\|_2}_{l_2 \text{ norm}} \leq r$ ↳ max norm hyper parameter

फरवरी February — रविवार Sunday 06

Here we do not add any regularisation term to the overall loss rather than for each training step

$$w \leftarrow \left(w \cdot \frac{r}{\|w\|_2} \right)$$ ↳ This is how we update the weights.

Conditions

1) if $\|w_{old}\|_2 \leq r$] No update of weight

$r = \|w\|_2$ ↳ No regularisation or scaling of the weights.

फरवरी February — सोमवार Monday 07

2) if $\|w\|_2 < r$ → weight update will be higher.

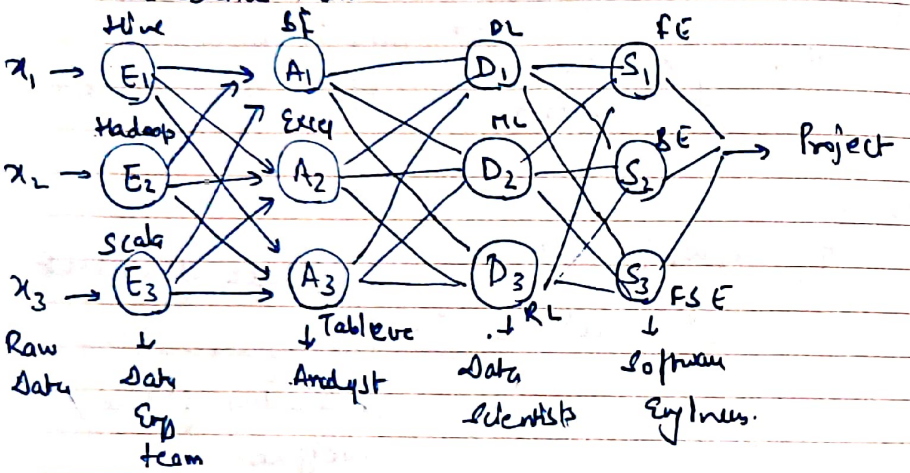
numerator > denominator

08 मंगलवार Tuesday — फरवरी February

Dropout

Most used and most successful regularisation technique

* Data science team



09 बुधवार Wednesday — फरवरी February

Let say in the Project cycle someone drops out from any one of the teams, so there work and expertise will not be available.

↓

Hence Project will come to halt as we have extreme dependency on each and every resources.

↓

Now remaining resources will try to learn something which people who left knew.

फरवरी February — गुरुवार Thursday 10

In dropout we do a simple technique of training the model with missing nodes. This will make the model much more impactful and effective.

Paper:- Dropout → Nitish Shrivastava et. al.

↓

Deep neural networks with a large number of parameters are very powerful machine learning systems. The problem remains of overfitting. Dropout is a technique to resolve it. It prevents the neural units from co-adapting with each other.

During training dropout trains a thin layer of exponential

फरवरी February — शुक्रवार Friday 11

networks which it tests jointly all the outputs at once and approximate avg predictions.

Its sort of an ensemble technique where we have different prediction branches and combine the output of all. This technique is not very advantageous as training multiple architectures is very hard and requires a lot of parameter tuning. Dropout is a better approach

Keeping probability = if $p = 0.7$ you want to keep 70% of the network active and remaining 30% as inactive

12 शनिवार Saturday — फरवरी February

Its better, if its closer to 1.

If there are n neural nets we can form maximum of 2^n thinned networks.

* Motivation :- Transfer of genes from sexual reproduction.

* Dropout is more generic in nature ~~can~~ and can be applied to any domain.

Mathematics

L → Total hidden layers. $l \in \{1, \dots, L\}$

13 रविवार Sunday — फरवरी February

$z^{(l)}$ → vector inputs into layer l

$y^{(l)}$ → vector output from layer l

$\{y^{(0)} = x$ is the input here$\}$

$w^{(l)}$, $b^{(l)}$ → weights and biases of layer l

$$\begin{cases} z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)} \\ y_i^{(l+1)} = f(z_i^{(l+1)}) \end{cases}$$

generic neural networks.

फरवरी February — सोमवार Monday 14

In case of dropouts

$r_j^{(l)} \sim$ Bernoulli (p) → Dropout step.

$\tilde{y}^{(l)} = r^{(l)} \otimes y^{(l)}$ → element wise product

$$\begin{cases} z_i^{(l+1)} = w_i^{(l+1)} \cdot \tilde{y}^{(l)} + b_i^{(l+1)} \\ y_i^{(l+1)} = f(z_i^{(l+1)}) \end{cases}$$

During testing - $w_{test}^{(l)} = p w^{(l)}$

फरवरी February — मंगलवार Tuesday 15

Drawback :-

① Increase the training time as it is more complex. It is the trade off between overfitting and training time.