② AdaGrad Optimization
(Adaptive Gradient)

$$r_t = r_{t-1} + (\nabla \theta_t)^2 \quad , \; r_0 \geq 0$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\delta + \sqrt{r_t}} \nabla \theta_t$$

↑ +ve number to avoid divide by 0.

⇒ As time progresses, r_t will get to a large value, hence, the moment does not happen.

③ 2014 AdaM Optimization (Adaptive Moments) — Adaptive + Momentums.

① Momentum →
$$s_t = \rho_1 s_{t-1} + (1-\rho_1)\nabla \theta_t \quad \text{— ①} \quad 0 \leq \rho_1 \leq 1$$

$$r_t = \rho_2 r_{t-1} + (1-\rho_2)(\nabla \theta_t)^2 \quad \text{— ②} \quad 0 \leq \rho_2 \leq 1$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\delta + \sqrt{\tilde{r}_t}} \cdot \tilde{s}_t$$

divide by zero, $\delta > 0$; ② Adaptive Gradient

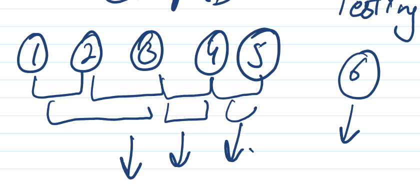
$$\tilde{r}_t = \frac{r_t}{1+\rho_2^t} , \quad \tilde{s}_t = \frac{s_t}{1-\rho_1^t}$$
→ ③ Bias Correction

REGULARIZATION

→ Generalized Network.

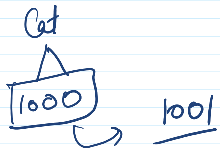
→ Training Sample (a distribution)
- Training
- Testing.

Noisy Training Samples — Testing



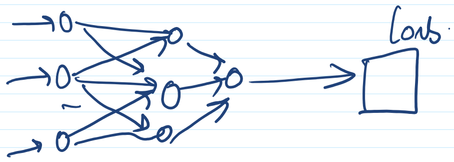
Domain-Shifting

ImageNet
1.3 Billion.

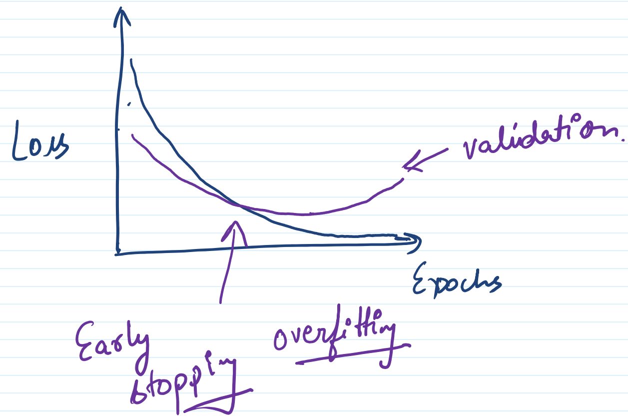


Dataset

- Training (80%)
  - Training (60%)
  - Validation (20%)
- Testing (20%)



100 it



① Early Stopping.

② Data Augmentation

10 class classification.

→ ① Cat → 100 images
→ ② Dog → 100 images
→ ③ | → 100 images
|

1000 images dataset → 1000 X 5 (80%)

→ Geometric Transformations:
Rotation, Translation, shear, scale, - - -

→ Photometric Transformations:
Noise, Blur, - - -

③ Adding regularization on weights.

$$\hat{L}(w) = L(w) + \frac{\beta}{2}\|w\|^2.$$

④ Noise injection :- to the input

MSE Loss

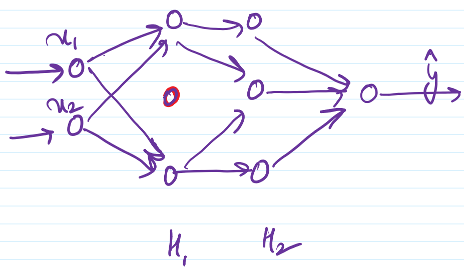
$$y_{i_{noisy}} = \sum w_i(x_i + \epsilon_i)$$

$$= \sum w_i x_i + \sum w_i \epsilon_i$$

$$= y + \underbrace{\sum w_i \epsilon_i}_{\text{regularization}}$$

Let. $\epsilon_i \sim (0, \sigma^2)$ ↑ Gaussian distribution.

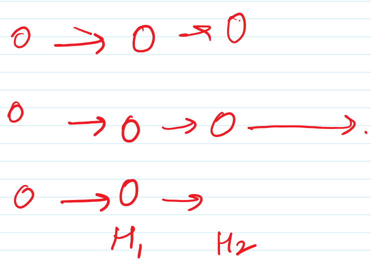
⑤ Drop out :-



'p' → probability of picking a neuron.

p = 0.5 50% of nodes of that hidden layer are dropped.

Mini-batch GD.



① If only some of the neurons take on the entire load of the task on hand, it can lead to overfitting.

② The rest of the neurons do not learn anything.

③ Typically, nodes are dropped only once for a mini-batch of sample.

4. Nodes are dropped with a probability of 'p'.

5. F/w & b/w pass are done only through active neurons.

6. At the time of testing, weights are multiplied by probability.

weights will reduce by factor 'p'.

Algorithm.	No. of steps in 1 epoch
Vanilla GD	1
Stochastic GD	N.
Minibatch.	N/B.

N → no. of training samples.

B → batch-size

Outputs

	Real Values (Regression)	Probability (Classification)
O/P activation	Linear	Sigmoid/ Softmax
Loss function	MSE	CCE.