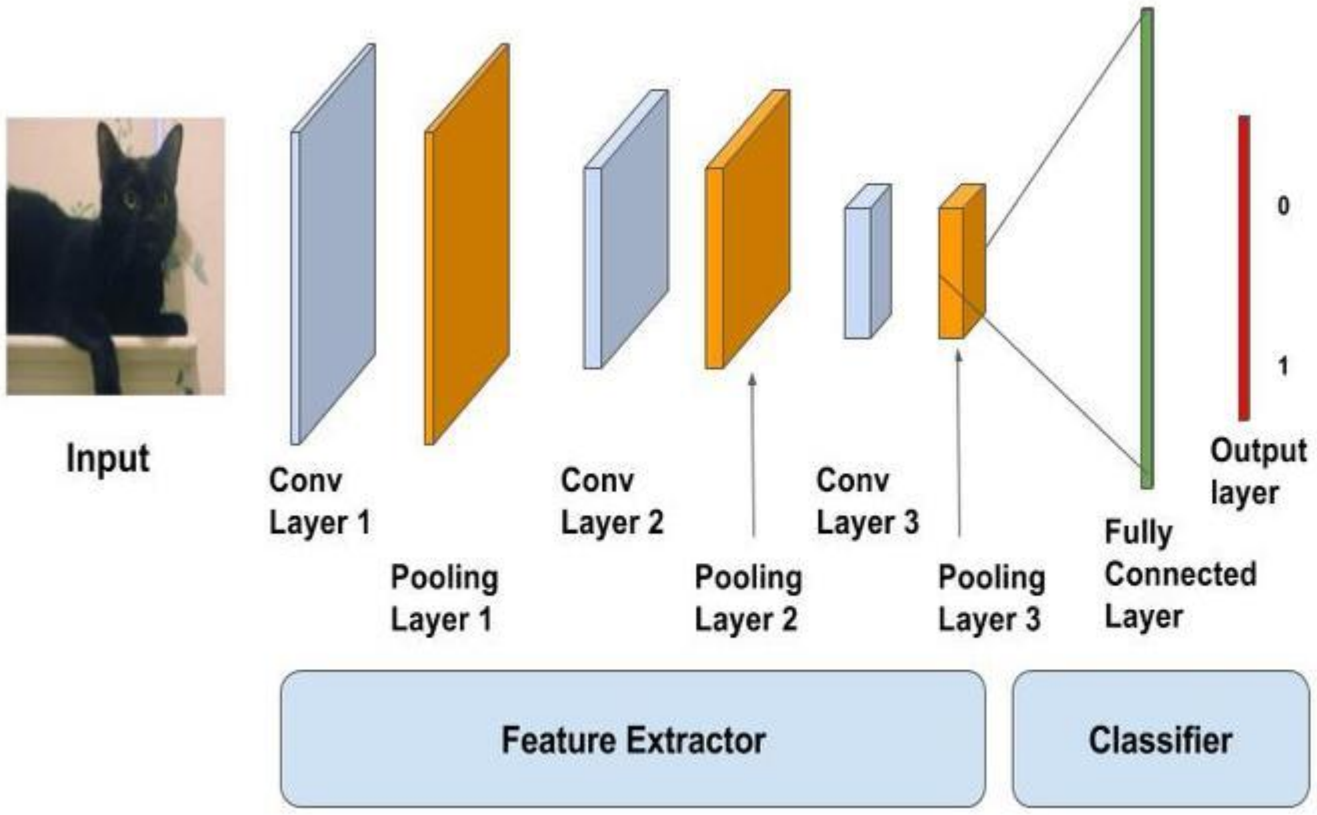
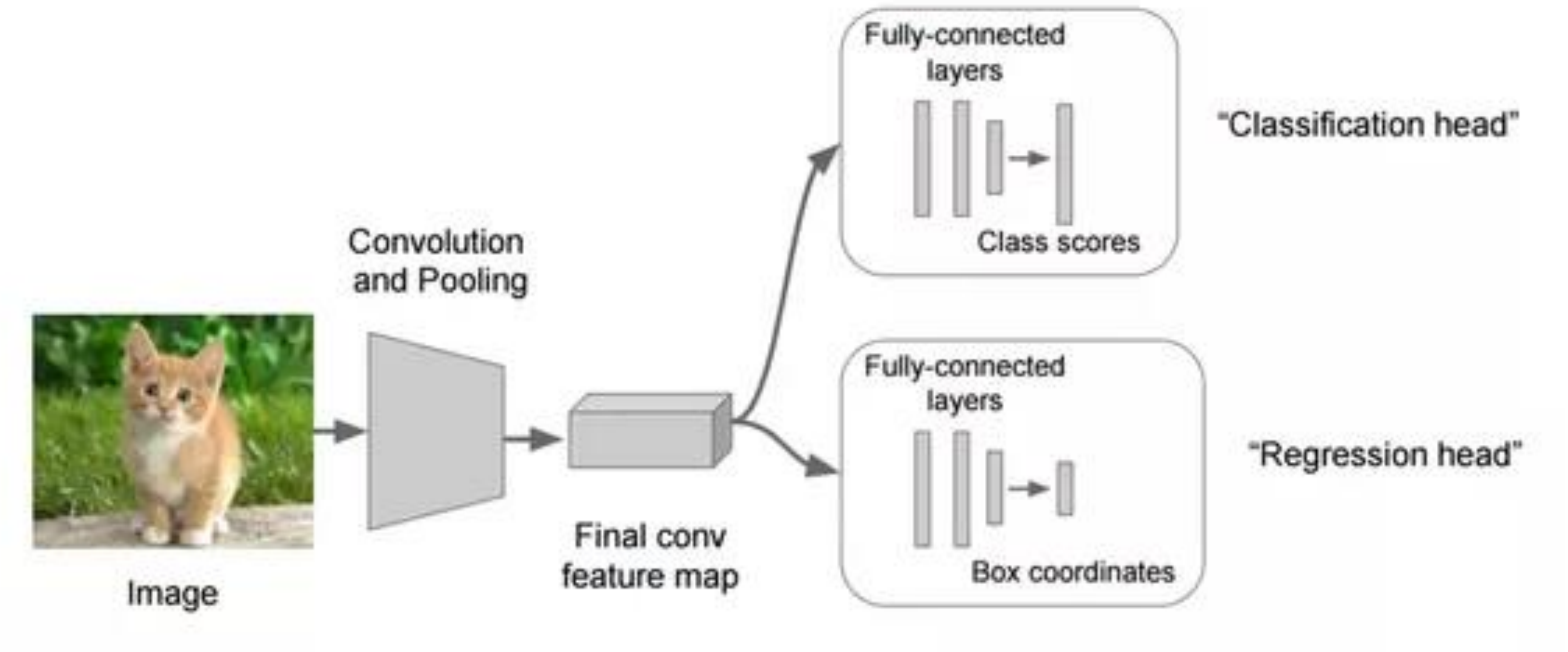
Closer look on the loss functions and optimizers

Biplab Banerjee

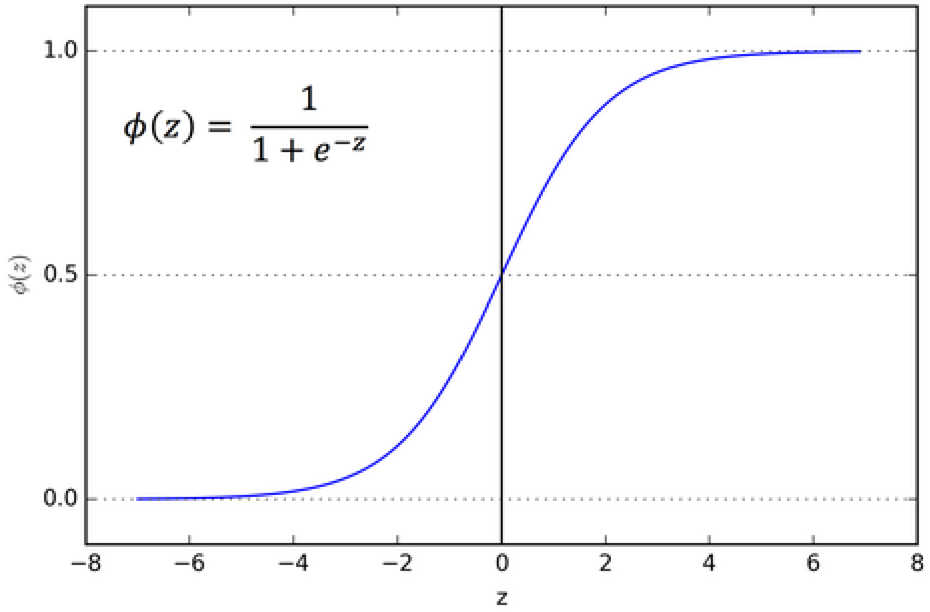
Vanilla CNN for classification

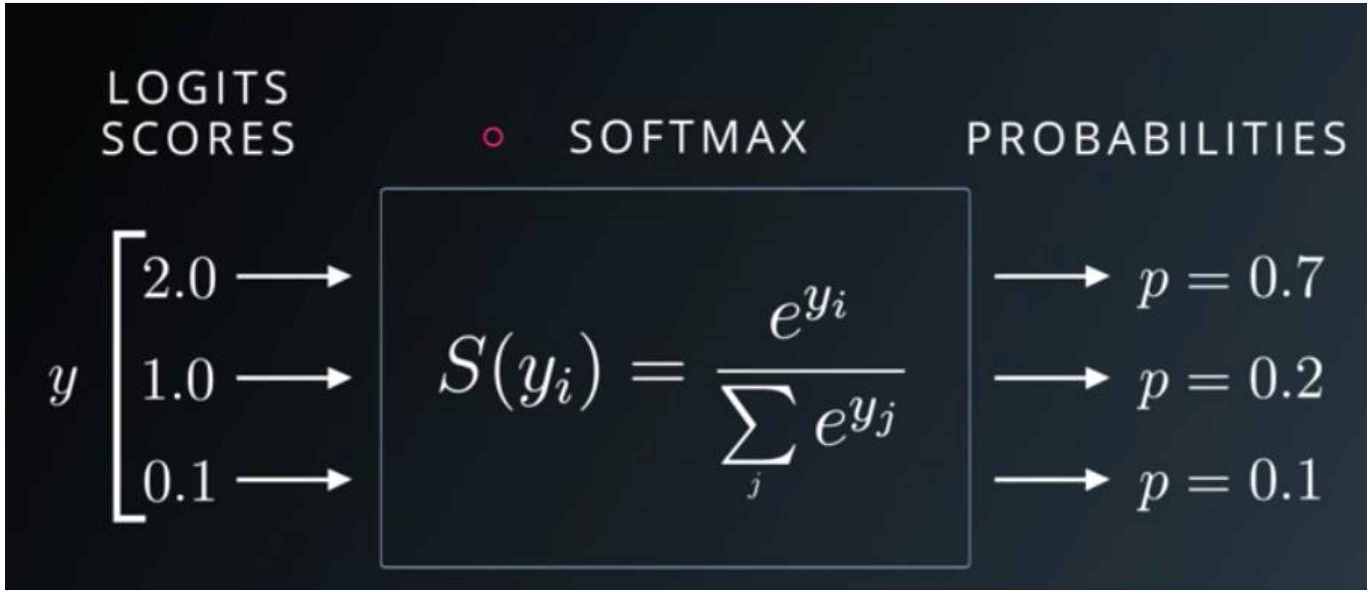


Vanilla CNN both for classification and regression



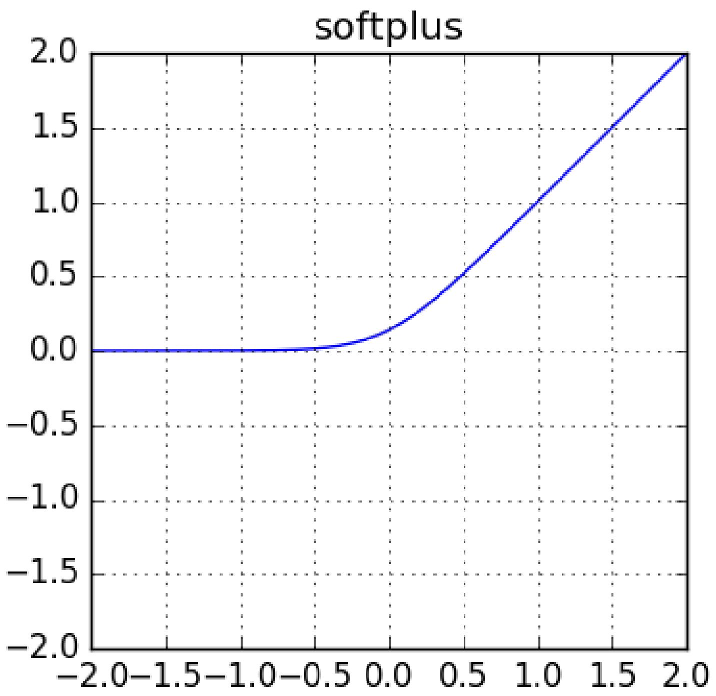
Activation functions for classification





- ✓ Sigmoid and softmax – convert logits into probabilities
- ✓ It is a monotonic function but the gradient is not!

Activation functions for classification

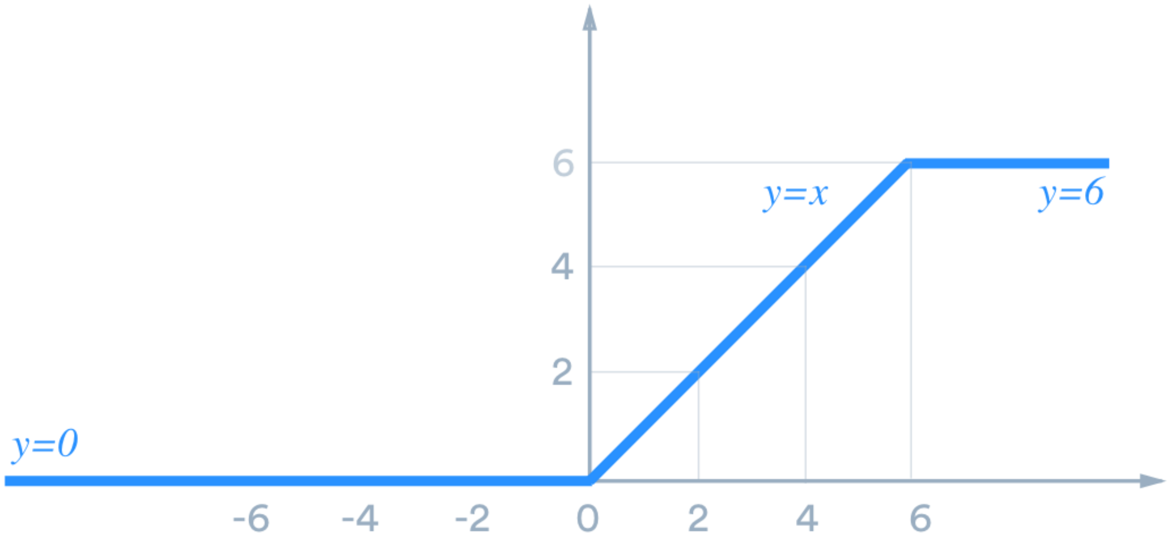


- ✓ What is the gradient of the Softplus function?
- ✓ What is the relation between Softmax and softplus?

$$f(x) = \ln(1+e^{x})$$

Activation function for regression

- Depends upon the continuous target value we want to estimate
- Based on the range, it could be sigmoid, tanh, or relu
- ReLU has the problem of “dying out”
- Some variants: Leaky ReLU, Parametric ReLU, Concatenated ReLU, ReLU6 etc.

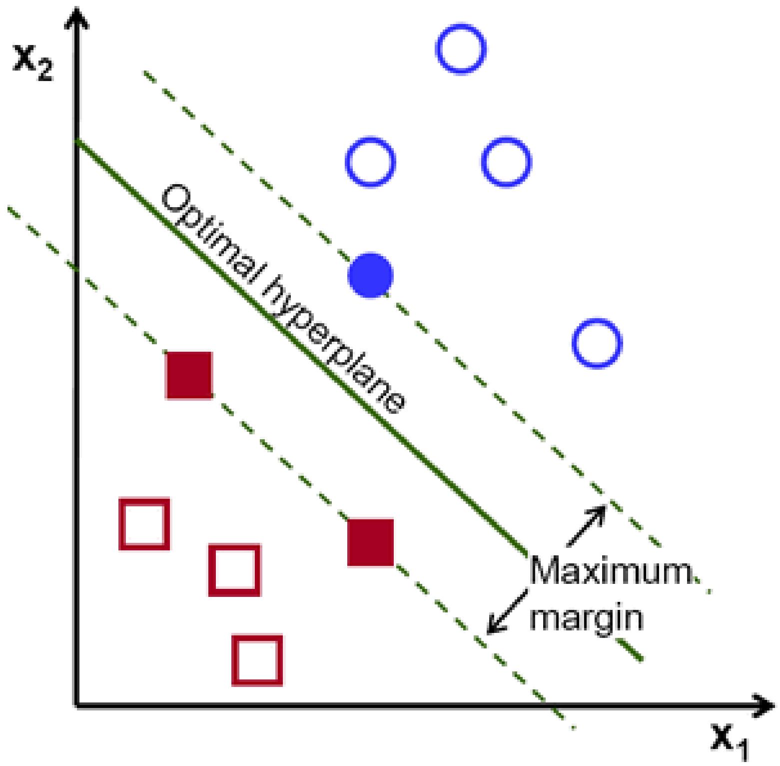


Loss functions

1. Regression Loss Functions
    1. Mean Squared Error Loss
    2. Mean Squared Logarithmic Error Loss
    3. Mean Absolute Error Loss
2. Binary Classification Loss Functions
    1. Binary Cross-Entropy
    2. Hinge Loss
    3. Squared Hinge Loss
3. Multi-Class Classification Loss Functions
    1. Multi-Class Cross-Entropy Loss
    2. Sparse Multiclass Cross-Entropy Loss
    3. Kullback Leibler Divergence Loss

$$L(y,\hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(log(y_i + 1) - log(\hat{y}_i + 1))^2$$

Hinge loss – margin based loss



$$L = \frac{1}{N} \sum_i \sum_{j \neq y_i} \left[\max\left(0, f(x_i; W)_j - f(x_i; W)_{y_i} + \Delta\right) \right] + \lambda \sum_k \sum_l W_{k,l}^2$$

- ✓ It is best for SVM type classifier
- ✓ \Delta controls the margin
- ✓ Let’s define ranking loss for image retrieval

Image retrieval

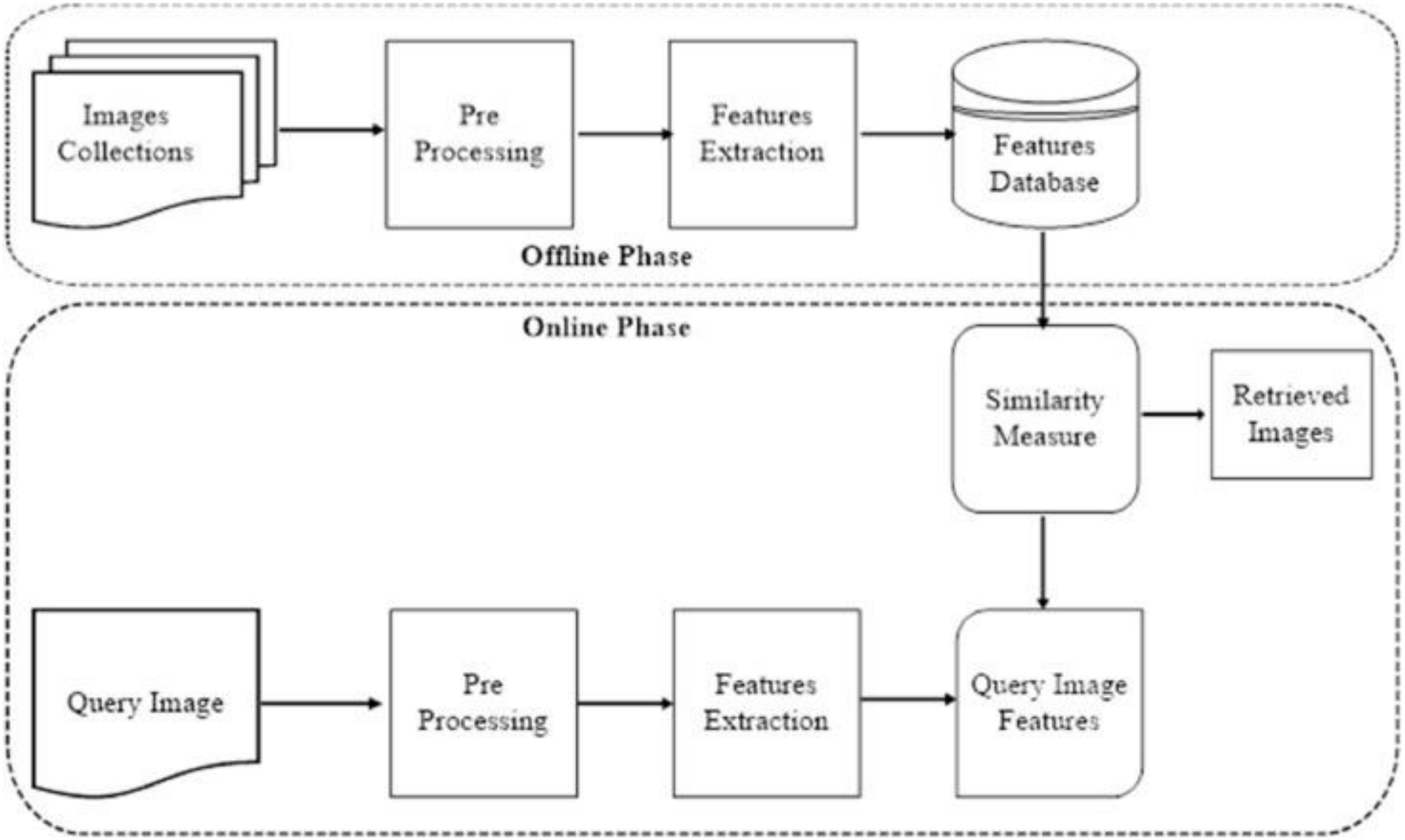
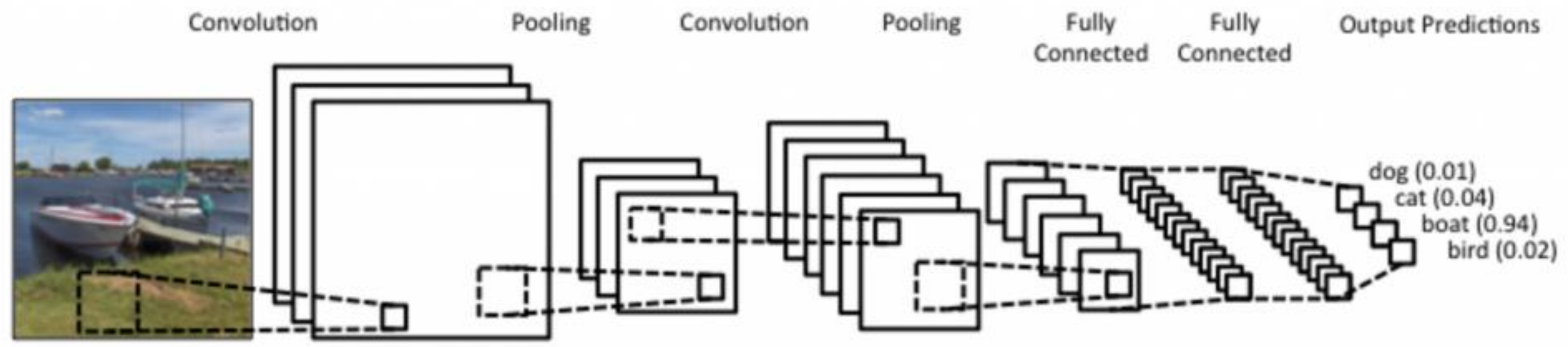


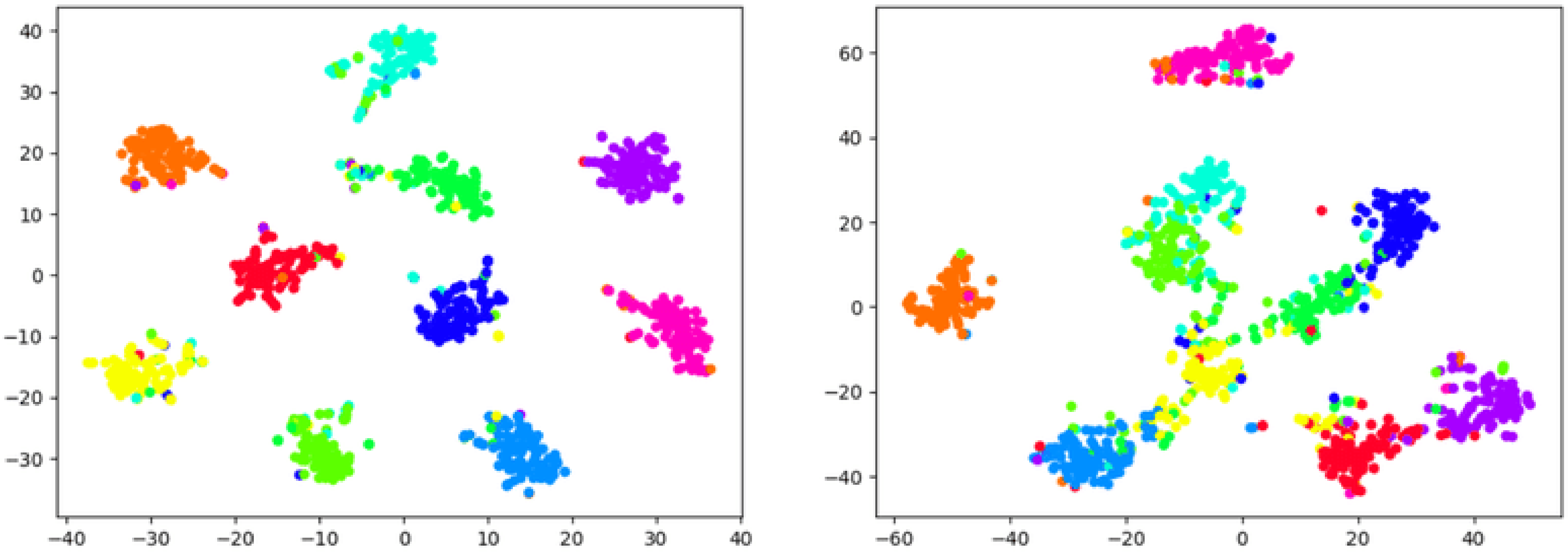
Image retrieval - what do we expect?

- Images of a given class should be compact in the feature space
- Images of different classes should be far apart
- For a given query image, there should be a ranking of all the possible retrieved images as per similarity

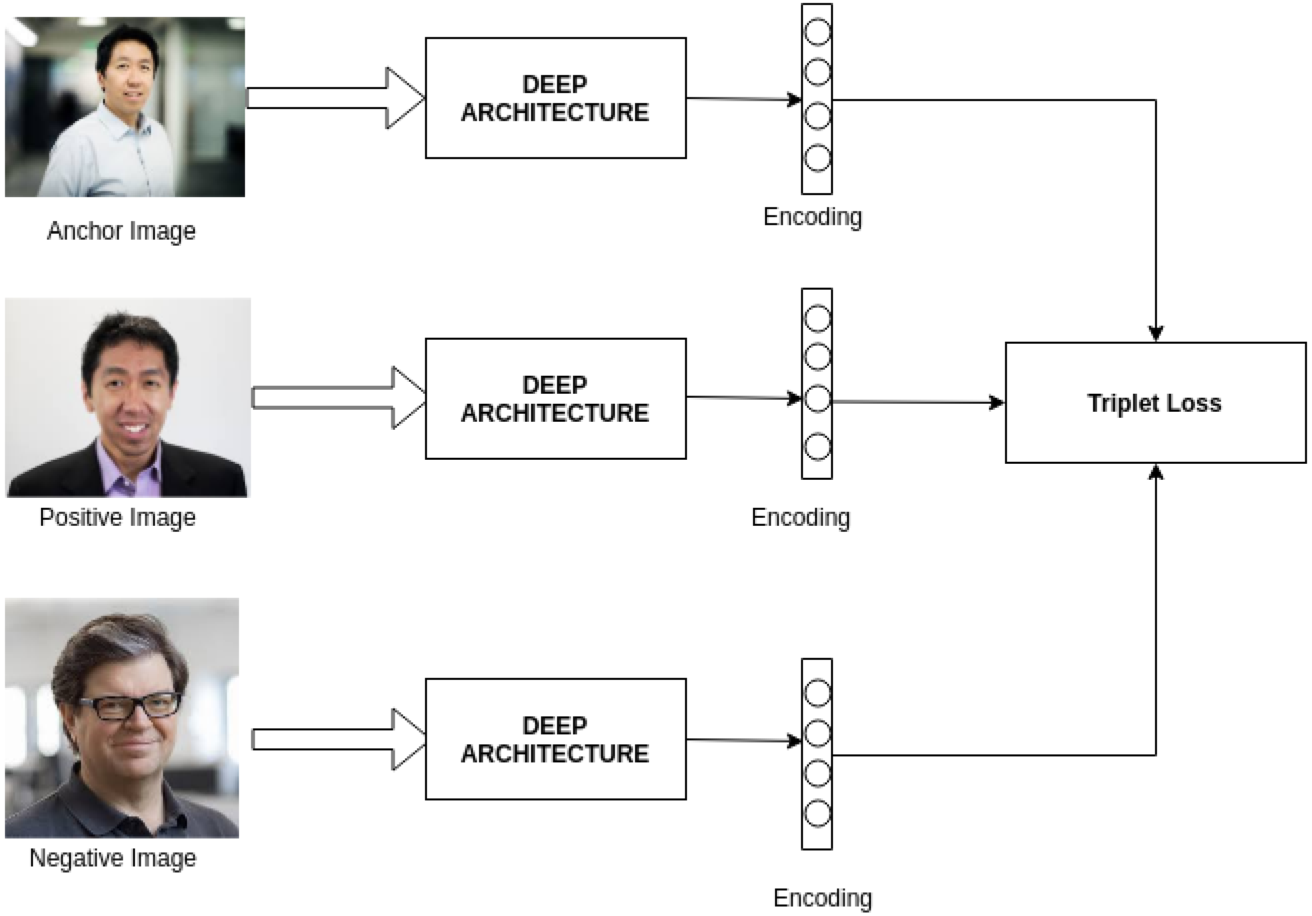
What is the feature vector corresponding to the image?



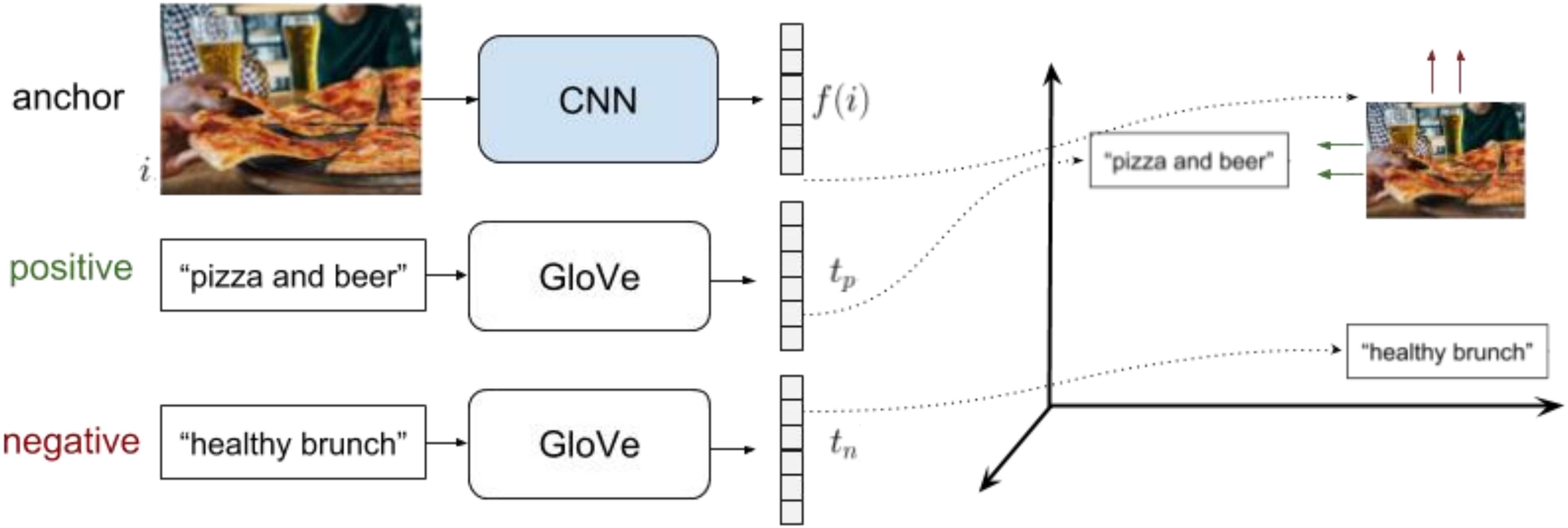
Which is a better feature space?



Content based image retrieval

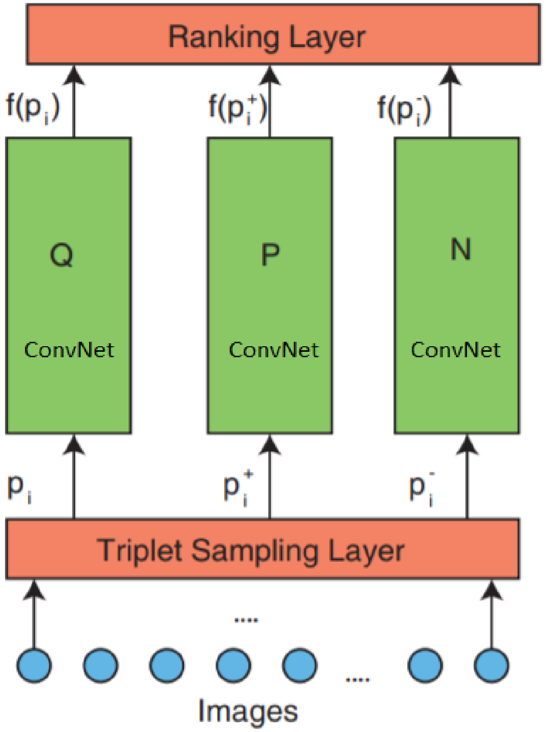


Text based image retrieval



Triplet network

What will be ranking loss here?



$$D(f(p_i), f(p_i^+)) < D(f(p_i), f(p_i^-)),$$
$$\forall p_i, p_i^+, p_i^- \text{ such that } r(p_i, p_i^+) > r(p_i, p_i^-)$$

$$l(p_i, p_i^+, p_i^-) = \max\{0, g + D(f(p_i), f(p_i^+)) - D(f(p_i), f(p_i^-))\}$$

Huber loss

Quadratic

$$L_\delta = \begin{cases} \frac{1}{2}\left(y - f(x)\right)^2, & if\ |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & otherwise \end{cases}$$

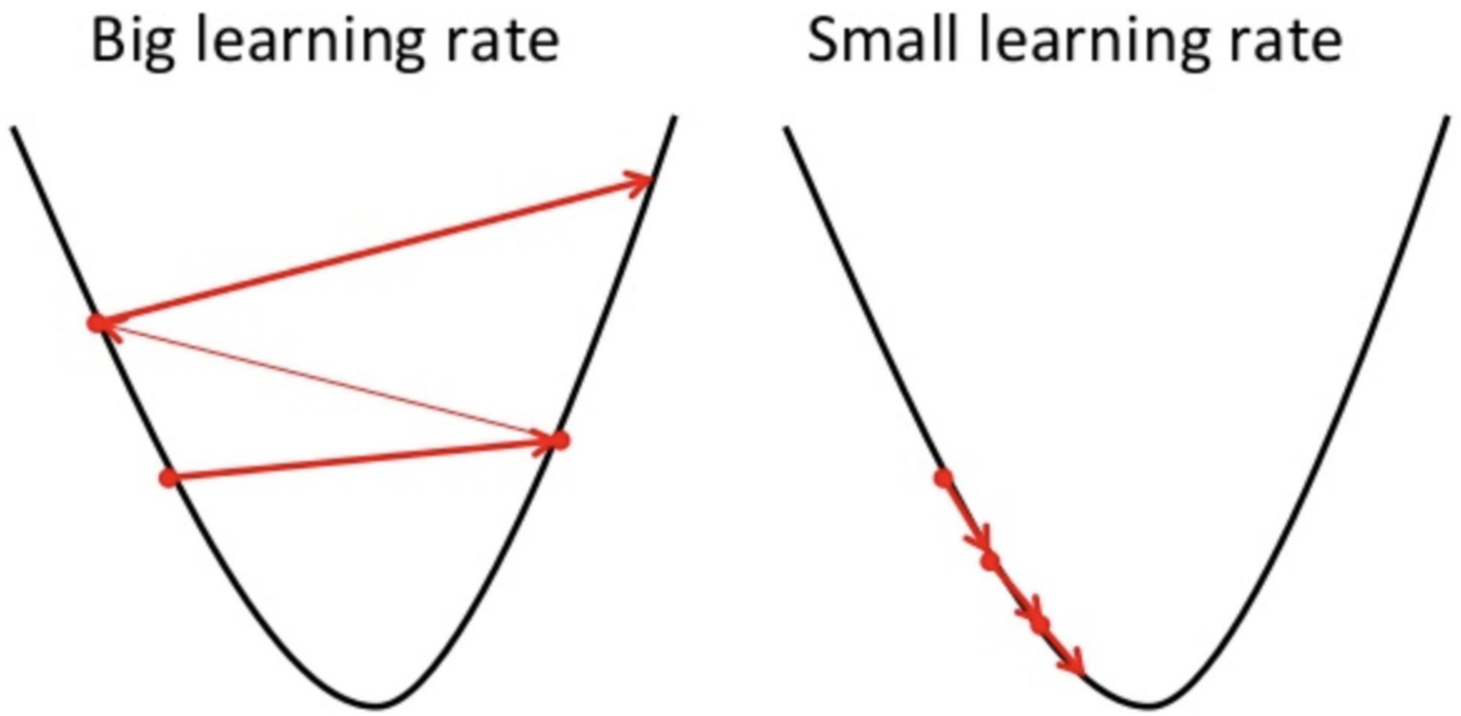
Linear

Combines the best of MSE and MAE

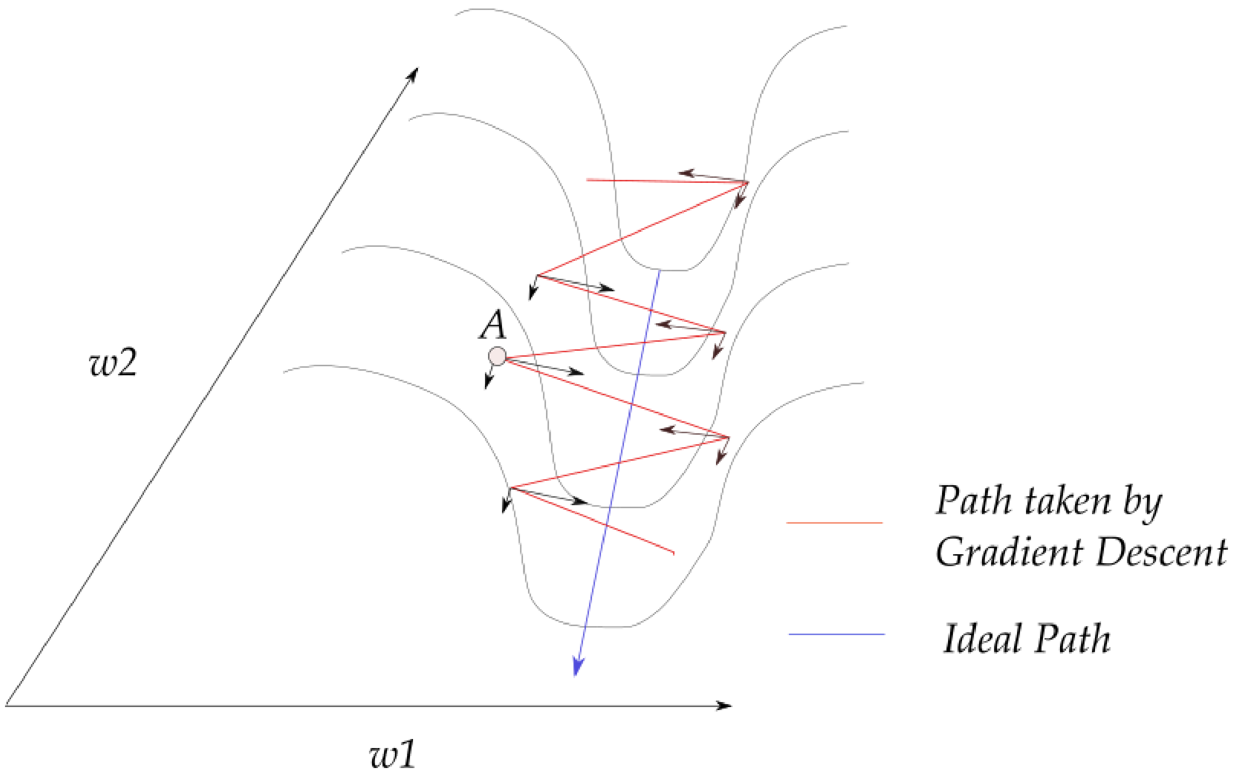
Some advanced optimizers

- Adam
- Adagrad
- Adadelta
- RMSProp
- SGD with momentum
- And many more…

Problem in selecting the learning rate



Momentum



Repeat Until Convergence {

$$\nu_j \leftarrow \eta * \nu_j - \alpha * \nabla_w \sum_{1}^{m} L_m(w)$$

$$\omega_j \leftarrow \nu_j + \omega_j$$

}

$$\boxed{\eta * \nu_j}$$

Coefficient of Momentum

Retained Gradient

Momentum

Example: 1. Let us consider the initial gradient is 0 and \nu=0.9

$$\nu_1 = -G_1$$

$$\nu_2 = -0.9 * G_1 - G_2$$

$$\nu_3 = -0.9 * (0.9 * G_1 - G_2) - G3 = -0.81 * (G_1) - (0.9) * G_2 - G_3$$

Essentially, there is exponential decay in the weights of the previous gradient terms!

RMSProp

$$For\ each\ Parameter\ w^j$$

$$(j\ subscript\ dropped\ for\ clarity)$$

Squared gradient

$$\nu_t = \rho\nu_{t-1} + (1-\rho) * g_t^2$$

$$\Delta\omega_t = -\frac{\eta}{\sqrt{\nu_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

η : *Initial Learning rate*

ν_t : *Exponential Average of squares of gradients*

g_t : *Gradient at time t along* ω^j

Adam

$$For\ each\ Parameter\ w^j$$

$$(j\ subscript\ dropped\ for\ clarity)$$

$$\nu_t = \beta_1 * \nu_{t-1} - (1 - \beta_1) * g_t$$

$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2$$

$$\Delta\omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

$\eta : Initial\ Learning\ rate$

$g_t : Gradient\ at\ time\ t\ along\ \omega^j$

$\nu_t : Exponential\ Average\ of\ gradients\ along\ \omega_j$

$s_t : Exponential\ Average\ of\ squares\ of\ gradients\ along\ \omega_j$

$\beta_1, \beta_2 : Hyperparameters$

Combines the effects of both RMSProp and Momentum.