DLCV

Week 3

Consider a simple perceptron f such that $f:R^4 \rightarrow R$ which uses the tanh function as its activation function. The input vector $X=[0.5,-0.7,0.9,0.3]$ and the corresponding weight vector $W=[0.6,1.5,-0.4,0.8]$. Compute the derivative of tanh(z), where z is the linear combination of the weights vector W and the input vector X. Ignore the bias term

Ans: z = W.X i.e z = -0.87

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\frac{d}{dz}\tanh(z) = 1 - \tanh^2(z)$$

The derivative of the tanh function at z=−0.87 is approximately **0.5086**.

Which of the following statements is **true** regarding the impact of dropout on the network's performance and the final learned weights?

A. Dropout during training causes the network to overfit because it randomly drops neurons, making the model too simple.

B. During training with dropout, the network adjusts the activation levels so that the outputs remain consistent when dropout is not applied during inference.

C. Dropout during training leads to a smaller final model size (fewer parameters) as some connections are permanently removed.

D. The final output during training (when dropout is applied) is multiplied by the dropout rate to maintain the expected output when dropout is not applied during inference.

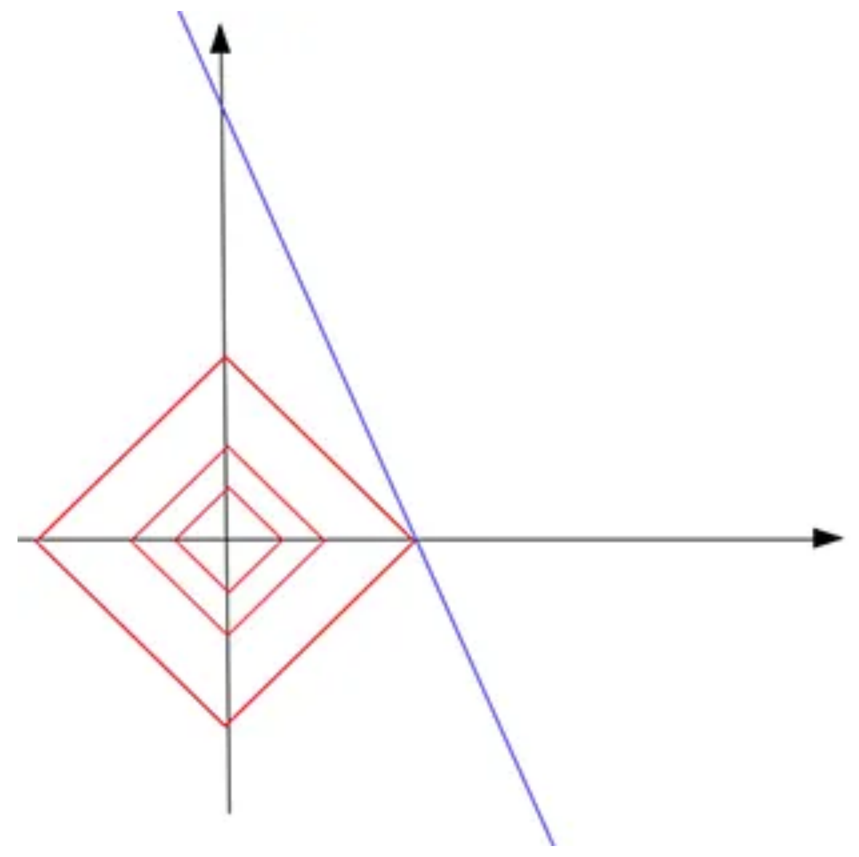
Ans: B. During training with dropout, the network adjusts the activations (by multiplying them by the inverse of the dropout rate) so that the output remains consistent when all neurons are active during inference. This adjustment ensures that the expected value of the activations remains the same, providing stability to the network's performance when dropout is no longer applied.

In the context of training a neural network, consider the following statements about L1 regularization, L2 regularization, and batch normalization:

1. Batch normalization cannot be used together with L1 and L2 regularization to improve the robustness and generalization of the model.
2. L1 regularization can lead to sparse weight matrices, which means many of the weights are zero.
3. Batch normalization normalizes the inputs of each layer by subtracting the batch mean and dividing by the batch standard deviation, slows down the training.
4. When using batch normalization, it is necessary to disable it during the inference phase.

Ans: 2, 4

L1 regularisation



Using RMSProp-based gradient descent, find the new value of parameter θ_{t+1}, given that the old value $\theta_t=1.8$, aggregated gradient $\Delta\theta_t=0.6$, gradient accumulation $r_{t-1}=0.4$, learning rate $\alpha=0.1$, decay rate $\rho=0.6$ and small constant $\delta=10^{-9}$. (Round decimal point till 3 places.)

Ans: 1.703

The formula for RMS prop is

1. Update the accumulated gradient:

$$r_t = \rho r_{t-1} + (1-\rho)(\Delta\theta_t)^2$$

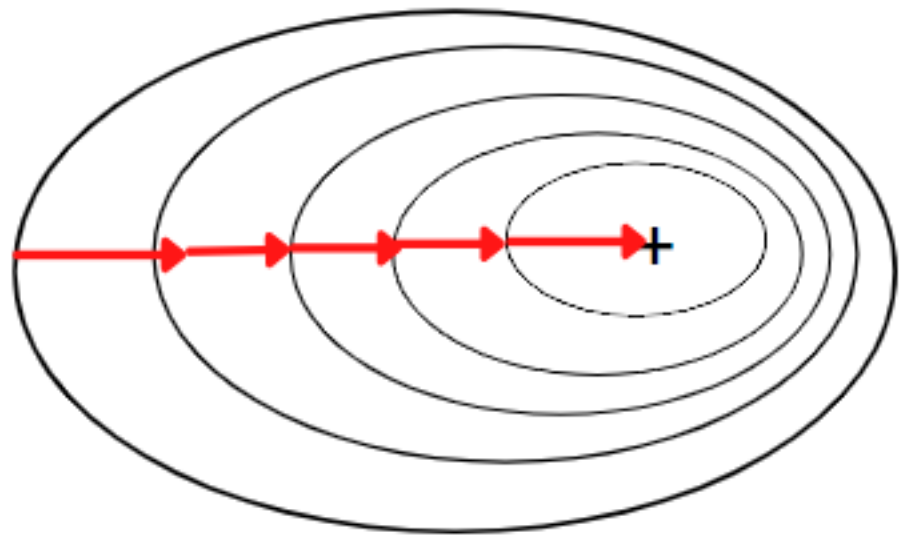
2. Update the parameter:

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

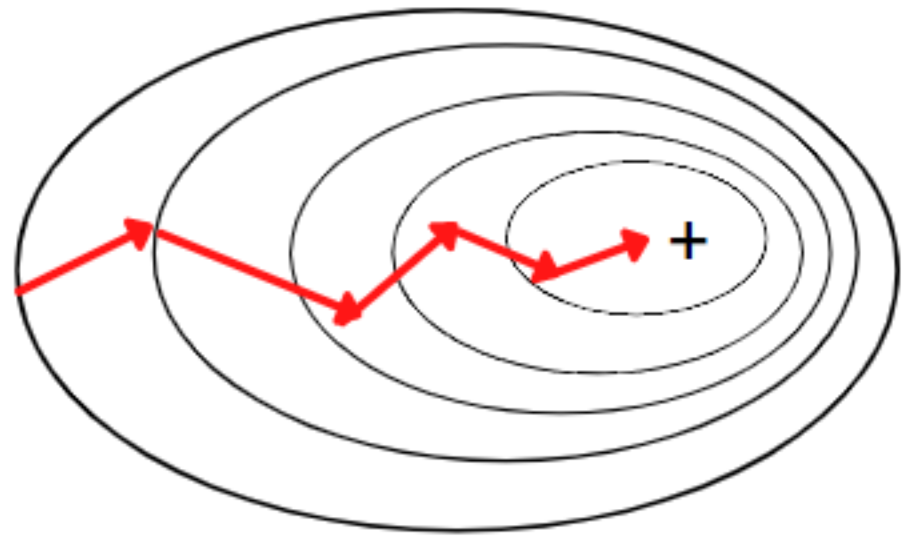
Consider a neural network being trained using different types of gradient descent methods. The following statements describe various aspects of these methods:

1. Batch Gradient Descent can lead to memory inefficiency when dealing with very large datasets
2. Stochastic Gradient Descent (SGD) often leads to a noisy path towards convergence but can help the model escape local minima more effectively.
3. Mini-Batch Gradient Descent is always faster in terms of computation time per epoch compared to Batch Gradient Descent.
4. Stochastic Gradient Descent guarantees faster convergence to the global minimum compared to Batch Gradient Descent and Mini-Batch Gradient Descent.
5. Using a smaller mini-batch size in Mini-Batch Gradient Descent generally increases the variance of the parameter updates.

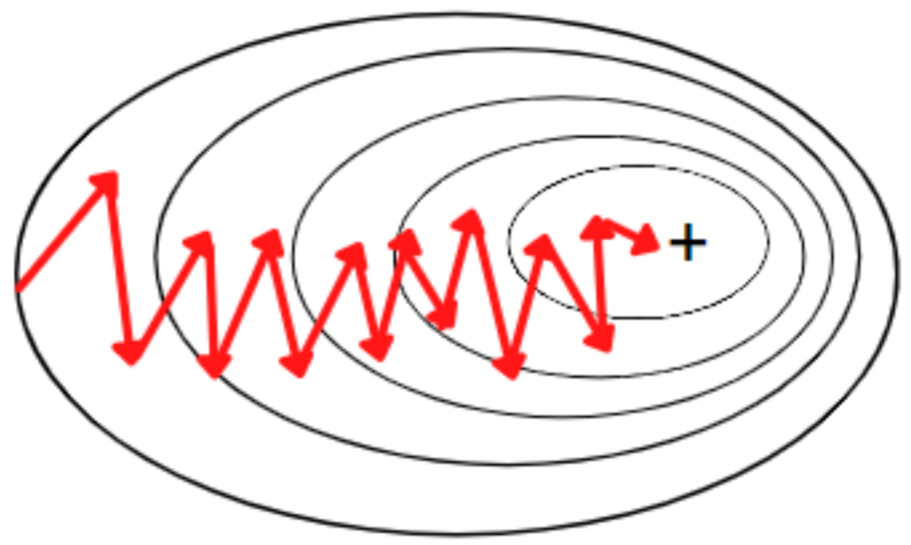
Batch Gradient Descent



Mini-Batch Gradient Descent



Stochastic Gradient Descent



You are training a neural network using Mixup augmentation. Consider two training samples (x1,y1) and (x2,y2) where:

x1=[0.2, 0.4, 0.6],y1=[1,0]

x2=[0.8, 0.7, 0.3],y2=[0,1]

Given the Mixup hyperparameter λ=0.3, calculate the Mixup augmented sample (xmix,ymix) xmix is [0.62, 0.61, 0.39]

ymix is [0.3, 0.7]

The mixup formula is as follows

$xmix = \lambda x1 + (1 - \lambda) x2$

$ymix = \lambda y1 + (1 - \lambda) y2$