Consider the RMSprop Optimization Algorithm (Choose all the correct answers) * 1 point

- [ ] The parameter p (rho) is a hyper-parameter and can take any value with no constraints
- [x] The parameter p (rho) is used to weight each parameter where the most recent gradients are weighted the most.
- [x] The previous gradients effect on the update step is exponentially decreased along with iterations.
- [ ] The global learning rate is the only factor that affects the update step size.

Learning Rate Decay Technique * 1 point

- ( ) Uses momentum to decrease the learning rate gradually.
- ( ) Decreases learning rate according to the size of the neural network
- ( ) Increases learning rate to help converging faster
- (•) Addresses the problem of the local minima overshooting

The AdaGrad algorithm (Choose all correct answers) * 1 point

- [x] Has large update steps at flat/saddle regions, small updates at steep regions
- [ ] Has small update steps at flat/saddle regions, large updates at steep regions
- [ ] Uses second derivative in the weight update
- [x] Can help in the problem of local minima overshooting

Convolution Neural Networks

CNNs Benefits are: (Choose all valid answers) * 1 point

- [ ] They can reduce overfitting with data.
- [x] Feature extracting for signals like images, these features help in reducing network size with more efficiency in learning.
- [ ] Normalizing image pixels
- [x] Parameter Sharing between different regions of images & Sparse Interactions.

Choose the valid sentences * 1 point

- [x] We use Dropout to solve the overfitting problem
- [x] Dropout kills neurons randomly while ReLUs kill only negative valued neurons
- [x] Dropout and ReLUs are similar in zeroing values
- [x] ReLUs can help solving the vanishing Gradient problem
- [ ] We use Conv layer to solve the underfitting problem

Derive a formula to compute the required padding for a same input-output dimension * 1 point

$$\frac{(S-1)I+F}{2}$$

- [ ] A

$$(S-1)I+F$$

- [ ] B

$$\frac{(S-1)I-S+F}{2}$$

- [x] C

$$\frac{(S-1)I+S-F}{2}$$

- [ ] D

Giving Input size of 224x224x3 with stride = 2 and padding= 1, Drive the output size using filter 4x4 with 64 kernels * 1 point

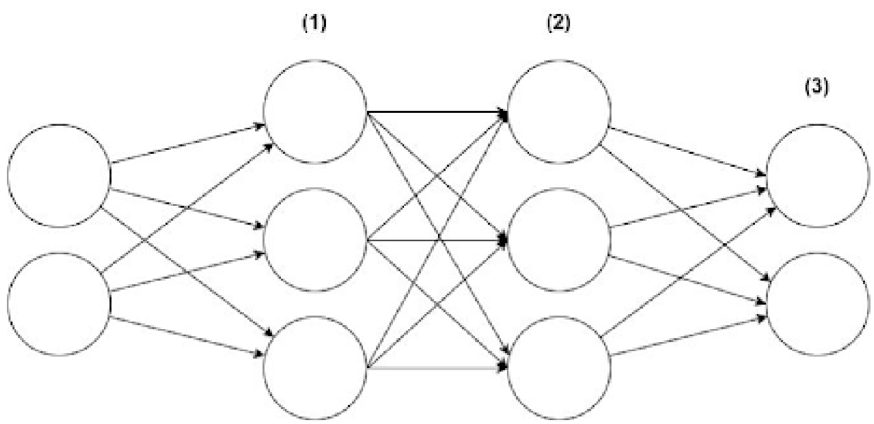
- [x] 112x112x64
- [ ] 112x112x3
- [ ] 224x224x64
- [ ] 112x224x64

Back-propagation

Practical Example

As an enthusiast about drones and quadcopters, you wanted to build a smart system in your drone that allows it to run in one of two modes (**Low power mode**, **High performance mode**), and this decision is made primarily on its 2d location so it can map low obstacle places with efficiency and high obstacle locations with safety.

So you turned to the tools you learnt in the Neural Networks Course and started building this feature … you knew instantly that a one hidden layer won't be enough and a one perceptron in the output will cause correlation between classes (you wanted if none of the modes are required, the model outputs small numbers and maintain the Normal mode of operation).



With these weights, a **ReLU** activation function in the hidden layers and a **Softmax** in the last layer you finally built your network and it was time to test it.

$$W^1 = \begin{bmatrix} W^1_{11} & W^1_{21} \\ W^1_{12} & W^1_{22} \\ W^1_{13} & W^1_{23} \end{bmatrix} = \begin{bmatrix} -2 & 3 \\ 1 & -0.5 \\ 0 & -1 \end{bmatrix}, \quad b^1 = \begin{bmatrix} b^1_{01} \\ b^1_{02} \\ b^1_{03} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$W^2 = \begin{bmatrix} W^2_{11} & W^2_{21} & W^2_{31} \\ W^2_{12} & W^2_{22} & W^2_{32} \\ W^2_{13} & W^2_{23} & W^2_{33} \end{bmatrix} = \begin{bmatrix} 0.5 & -1 & 4 \\ 0.1 & 2 & -5 \\ -6 & 0 & 3 \end{bmatrix}, \quad b^2 = \begin{bmatrix} b^2_{01} \\ b^2_{02} \\ b^2_{03} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$W^3 = \begin{bmatrix} W^3_{11} & W^3_{21} & W^3_{31} \\ W^3_{12} & W^3_{22} & W^3_{32} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0.5 \\ 5 & -0.1 & 0.5 \end{bmatrix}, \quad b^3 = \begin{bmatrix} b^3_{01} \\ b^3_{02} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

What will be the loss if the input was [6, 4] and the correct class was 2 ? * 1 point

- [ ] [0.73, 0.268]
- [x] 1.31
- [ ] [0.313, 1.313]
- [ ] 0.313
- [ ] I hate drones

Continuing with the results from the previous question * 1 point

$$Compute\ \frac{\partial L}{\partial W_{22}^{2}}$$

- [x] 0.292
- [ ] 2.92
- [ ] 0.0
- [ ] -17.54

What will be the second bias value in the first layer after the update (learning rate is 0.5) ? * 1 point

$$b_{02}^{1} = b_{02}^{1} - \alpha \frac{\partial L}{\partial b_{02}^{1}}$$

- (x) -0.073
- ( ) 0
- ( ) 0.146
- ( ) 0.182