

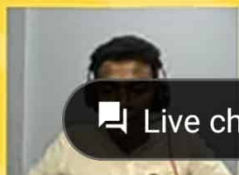
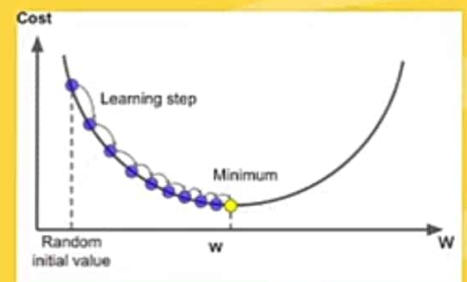
GRADIENT DESCENT

Optimizer- To minimize the loss function or error term

We start with a random point on the function and move in the negative direction of the gradient of the function to reach the local/global minima

Gradient descent algorithm needs,

- we require a Cost function that needs to be minimized,
- the number of iterations,
- a learning rate to determine the step size at each iteration while moving towards the minimum,
- partial derivatives for weight & bias to update the parameters at each iteration,
- and a prediction function

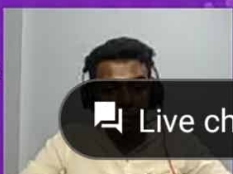


DEEP LEARNING TERMINOLOGY - 1



Learning Rate

- Step size at each iteration while moving towards the minimum
- The amount that the weights are updated during training is referred to as the Learning Rate'
- Positive value, often in the range between 0.0 and 1.0
- The learning rate controls how quickly the model is adapted to the problem
- Smaller learning rates require more training epochs given the smaller changes made to the weights each update
- Larger learning rates result in rapid changes and require training epochs



DEEP LEARNING TERMINOLOGY - 2

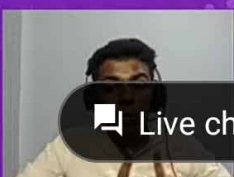


Hyper parameter

- A hyper parameter is a parameter whose value is set before the learning process begins
- It determines how a network is trained and the structure of the network

Types

- Dropout
- Activation function
- Learning Rate
- Momentum
- Number of epochs
- Batch size



DEEP LEARNING TERMINOLOGY - 3



Bias, Variance

- **Bias** – Assumption made by model to make function easier to learn
- If average predicted value are far from actual value then bias is said to be high
- **Variance** – Variability of model prediction for a given data point which tells the spread of data called variance of the model
- Model with high variance very flexible to training data but it cannot fit accurately on test data



GradientDescent_DL.ipynb

☆

File Edit View Insert Runtime Tools Help Last saved at 17:26

+ Code + Text

RAM Disk

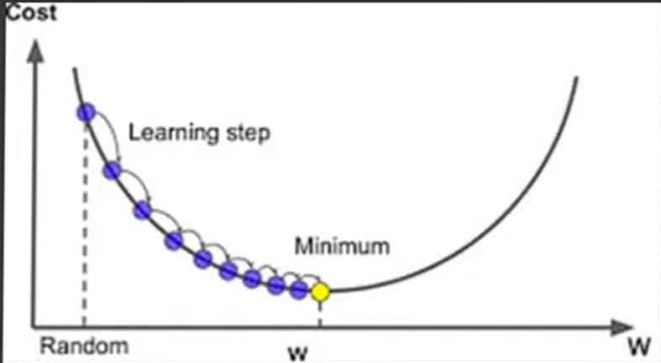
Editing

+ Code + Text

↑ ↓ ↻ ⌨ 📄 🗑 ⋮

Optimization - Gradient Descent

Gradient of Function



Live ch

GradientDescent_DL.ipynb - Colab

colab.research.google.com/drive/1lgxjFwajyyZB2LW0HHaE4O-Q-1rMZyFw#scrollTo=iHX2LPNbc2bl

GradientDescent_DL.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

```
[17] functionGradient = lambda x: 2*x*x

learningRate = 0.01
precisionVar = 0.000001
stepSize = 1
maxEpoch = 10000
iterCounter = 0
currentX = 3
while stepSize > precisionVar and iterCounter < maxEpoch:
    prev_x = currentX
    currentX = currentX - learningRate * functionGradient(prev_x)
    stepSize = abs(currentX - prev_x)
    iterCounter = iterCounter+1
    print("Iteration: {0}, Value: {1}".format(iterCounter,currentX))

print("Minimum occurs at {0} & in the Iteration {1}".format(currentX,iterCounter))
```

0s completed at 19:23

Live