NPTEL ONLINE CERTIFICATION COURSES



# Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

## Topic

Lecture 41: Popular CNN Models V

CONCEPTS COVERED

## Concepts Covered:

- CNN
  - AlexNet
  - VGG Net
  - Transfer Learning
  - Challenges in Deep Learning
  - GoogLeNet
  - ResNet
  - etc.

# Challenges

- ❑ Deep learning is data hungry.
- ❑ Overfitting or lack of generalization.
- ❑ Vanishing/Exploding Gradient Problem.
- ❑ Appropriate Learning Rate.
- ❑ Covariate Shift.
- ❑ Effective training.

# Vanishing Gradient Problem

$$X \xrightarrow[W_1]{} f_1 \xrightarrow[W_2]{} f_2 \xrightarrow[W_3]{} f_3 \xrightarrow[W_4]{} f_4 \longrightarrow O$$

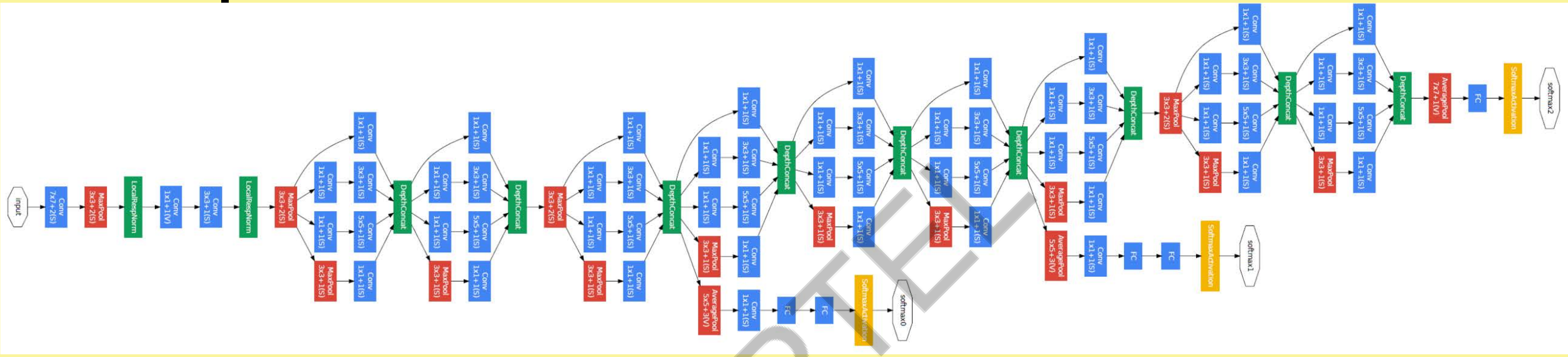
$$\frac{\partial O}{\partial W_1} = X \,.\, f_1' . W_2 . f_2' . W_3 . f_3' . W_4 . f_4'$$

# Vanishing Gradient Problem

- Choice of activation function: ReLU instead of Sigmoid.
- Appropriate initialization of weights.
- Intelligent Back Propagation Learning Algorithm.

# GoogLeNet
# ILSVRC 2014 Winner

# GoogLeNet





❖22 Layers with parameters

❖27 Layer including Maxpool layers

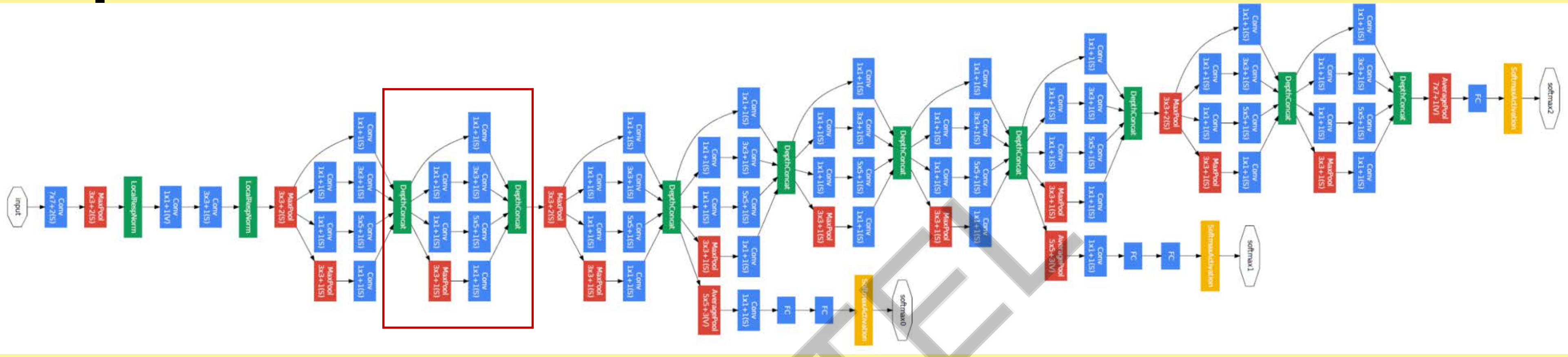
Convolution Layer

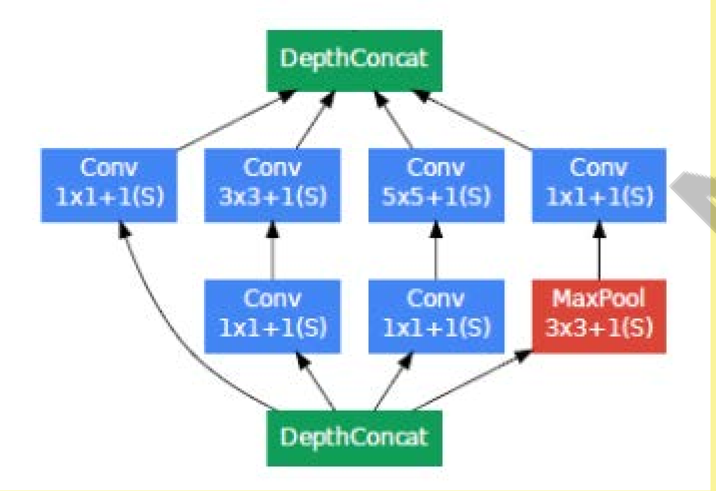
Maxpool Layer

Feature Concatenation

Softmax Layer

# GoogLeNet



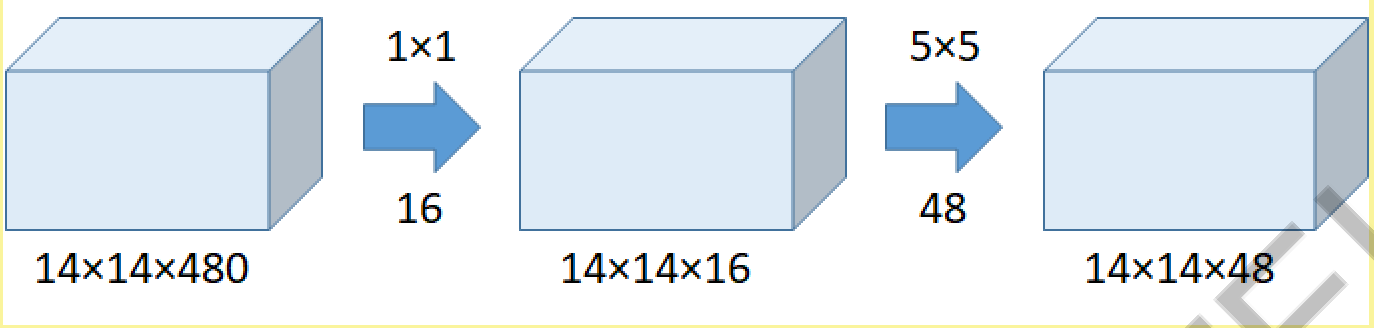



Inception Module

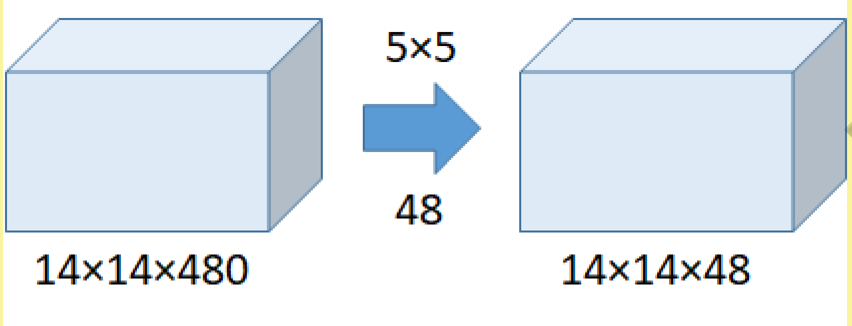
# Inception Module

- Computing 1×1, 3×3, and 5×5 convolutions within the same module of the network.
- Covers a bigger area, at the same time preserves fine resolution for small information on the images.
- Use different convolution kernels of different sizes in parallel from the most accurate detailing (1x1) to a bigger one (5x5).
- 1x1 convolution also reduces computation.

# Inception Module




Number of operations for 1×1 = (14×14×16)×(1×1×480) = 1.5M
Number of operations for 5×5 = (14×14×48)×(5×5×16) = 3.8M
Total number of operations = 1.5M + 3.8M = 5.3M



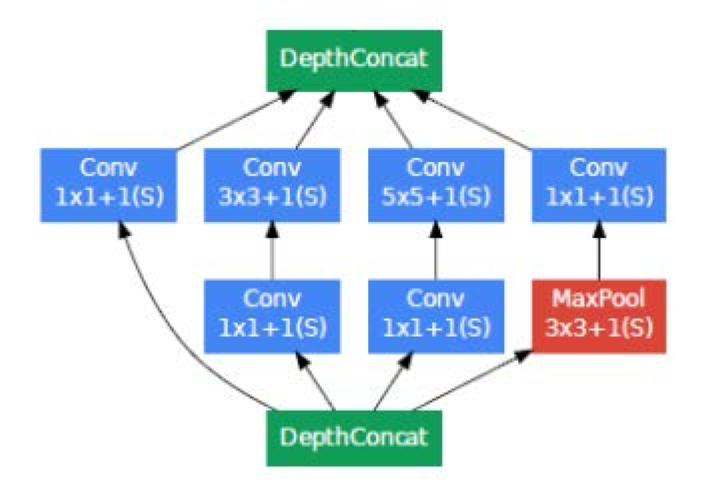

Number of operations = (14×14×48)×(5×5×480) = 112.9M



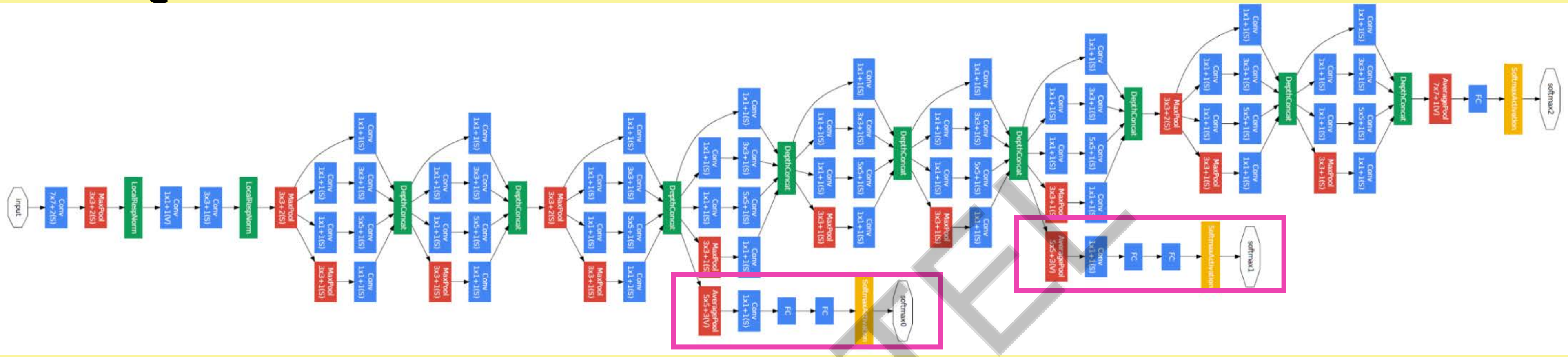
https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7

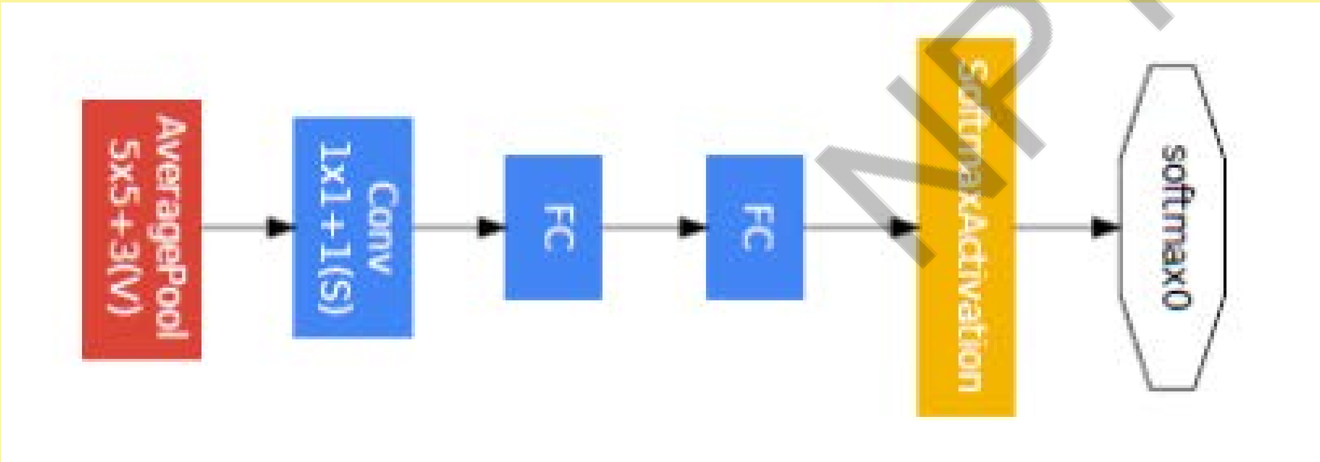
# Inception Module

- Outputs of these filters are then stacked along the channel dimension.
- Multi-level feature extractor.
- There are 9 such inception modules.
- Top-5 error rate of less than 7 %.



# GoogLeNet






Auxiliary Classifier

# Auxiliary Classifier

- Due to large depth of the network, ability to propagate gradient back through all the layers was a concern.
- Auxiliary Classifiers are smaller CNNs put on top of middle Inception modules.
- Addition of auxiliary classifiers in the middle exploits the discriminative power of the features produced by the layers in the middle.

# Auxiliary Classifier

- During training, loss of Auxiliary classifiers are added to the total loss of the network.
- Losses from Auxiliary classifiers were weighted by 0.3.
- Auxiliary classifiers are discarded at Inference time.


NPTEL ONLINE CERTIFICATION COURSES

Thank you

# NPTEL ONLINE CERTIFICATION COURSES

## Course Name: Deep Learning


Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur




## Topic

### Lecture 42: Popular CNN Models VI

# CONCEPTS COVERED

Concepts Covered:

- CNN
  - Challenges in Deep Learning
  - GoogLeNet
  - ResNet
  - Momentum Optimizer

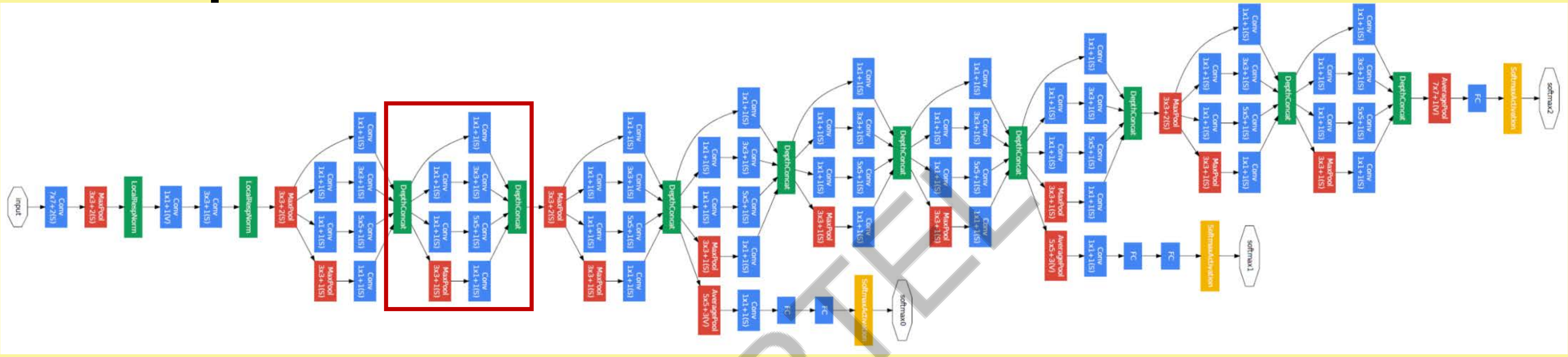
# Challenges

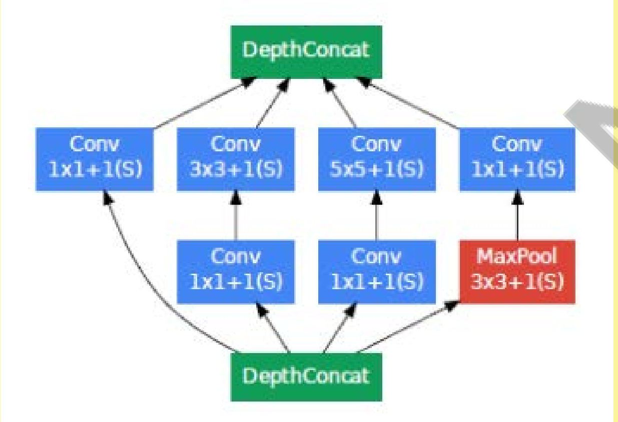
- Deep learning is data hungry.
- Overfitting or lack of generalization.
- Vanishing/Exploding Gradient Problem.
- Appropriate Learning Rate.
- Covariate Shift.
- Effective training.

# Vanishing Gradient Problem

- ❑ Choice of activation function: ReLU instead of Sigmoid.
- ❑ Appropriate initialization of weights.
- ❑ Intelligent Back Propagation Learning Algorithm.

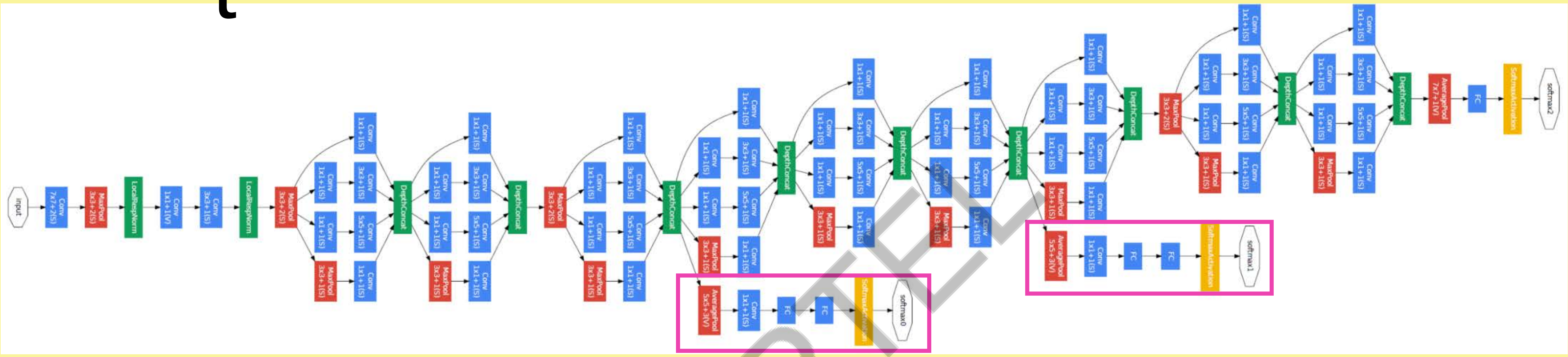
# GoogLeNet

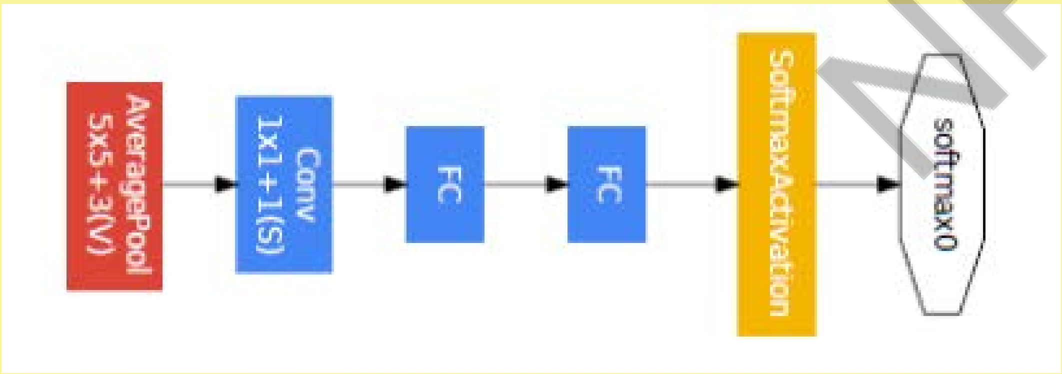





Inception Module

# GoogLeNet



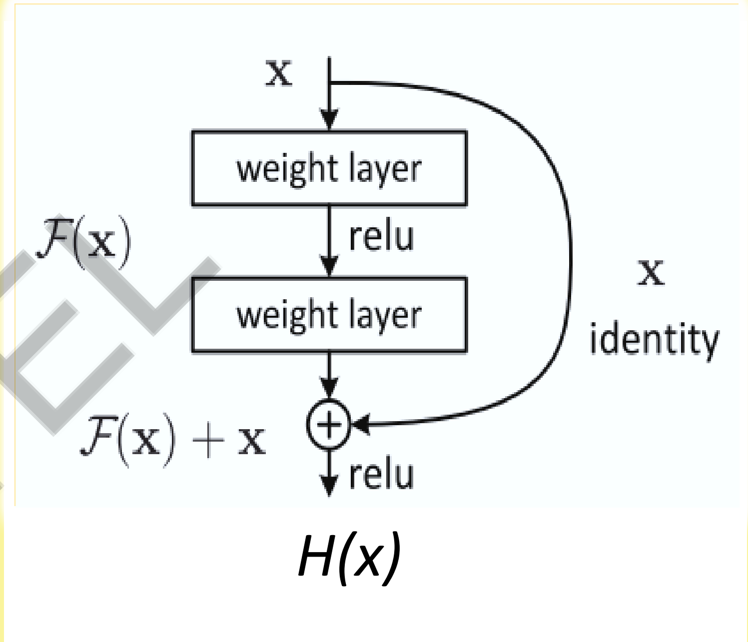



Auxiliary Classifier

# ResNet

# ResNet

- Core idea is: introduction of Skip Connection/ Identity Shortcut Connection that skips one or more layers.
- Stacking layers should not degrade performance compared to its shallow counterpart.
- Weight layer learns *F(x)=H(x)-x*

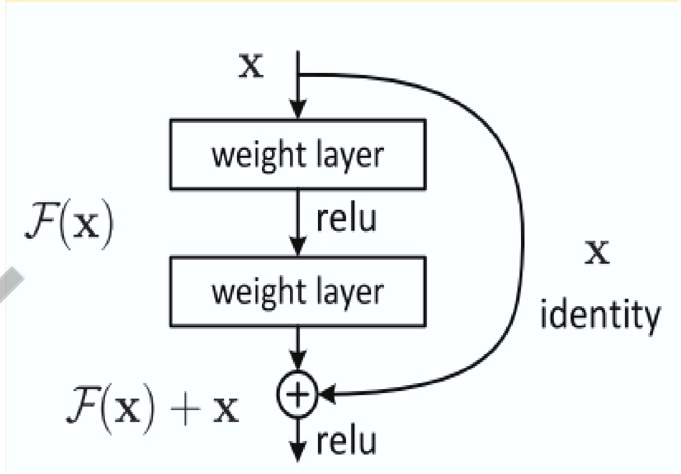





https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

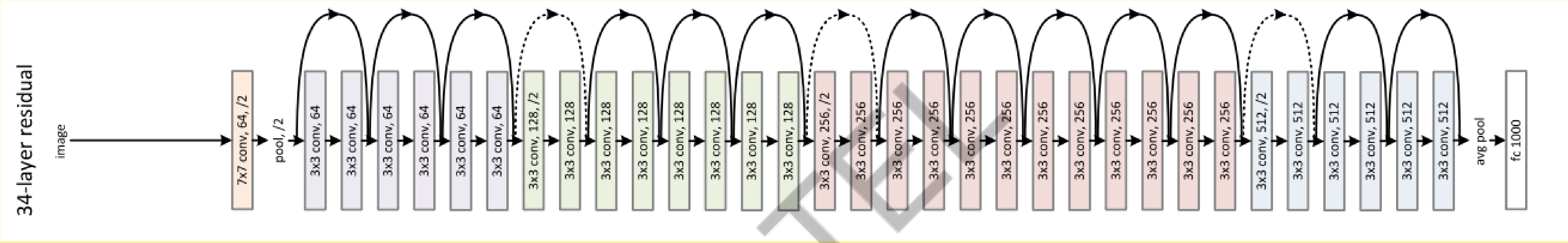
# ResNet

- By stacking identity mappings the resultant deep network should give at least same performance as its shallow counterpart.
- Deeper network should not give higher training error than shallow network.
- During learning the gradient can flow to any earlier network through shortcut connections alleviating vanishing gradient problem.




*H(x)*

# ResNet





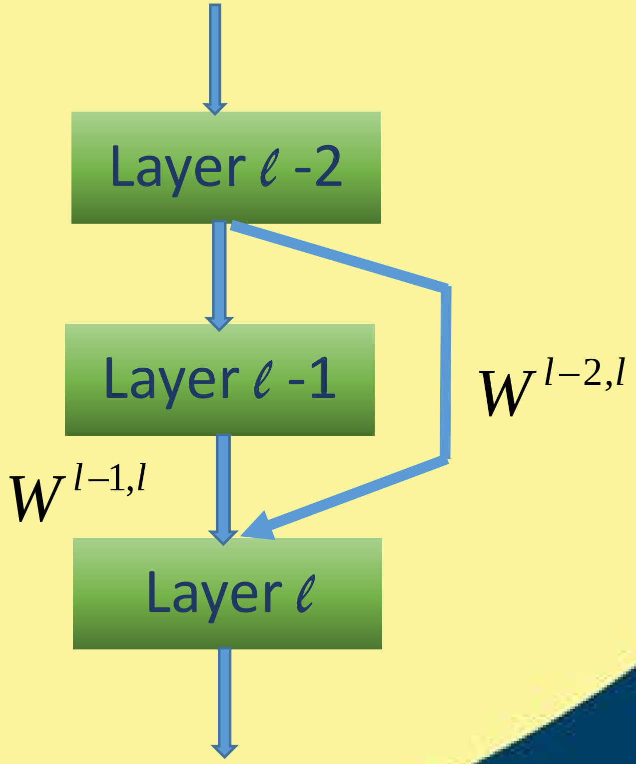

https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

# ResNet

Forward flow:

$$a^l = f(W^{l-1,l}.a^{l-1} + b^l + W^{l-2,l}.a^{l-2})$$
$$= f(Z^l + W^{l-2,l}.a^{l-2})$$

$$a^l = f(Z^l + a^{l-2}) \qquad \text{if same dimension}$$

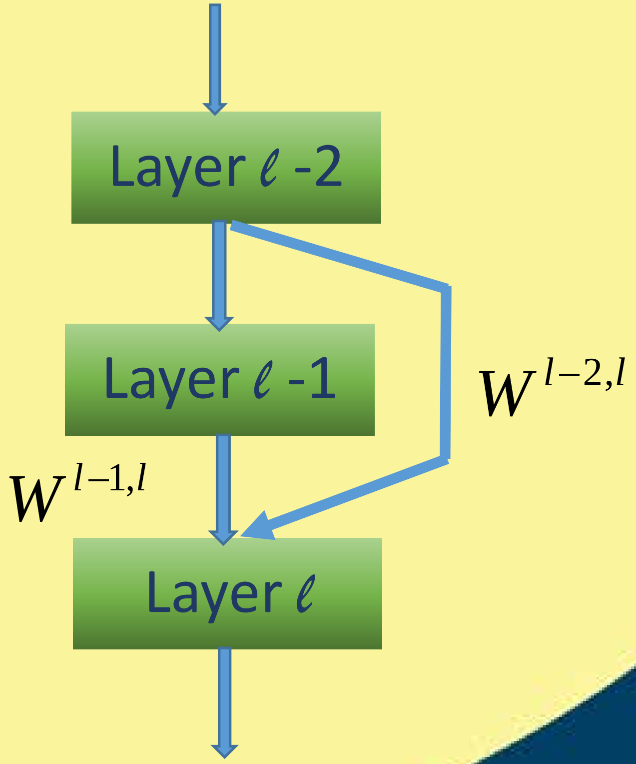


# ResNet

## Backward Propagation:

$$\nabla W^{l-1,l} = -a^{l-1}.\delta^{l} \quad \text{normal path}$$

$$\nabla W^{l-2,l} = -a^{l-2}.\delta^{l} \quad \text{skip path}$$




If the skip path has fixed weights, identity matrix, then they are not updated.

# Challenges

- ❑ Deep learning is data hungry.
- ❑ Overfitting or lack of generalization.
- ❑ Vanishing/Exploding Gradient Problem.
- ❑ Appropriate Learning Rate.
- ❑ Covariate Shift.
- ❑ Effective training.

# Optimizing Gradient Descent

# Gradient Descent Challenges

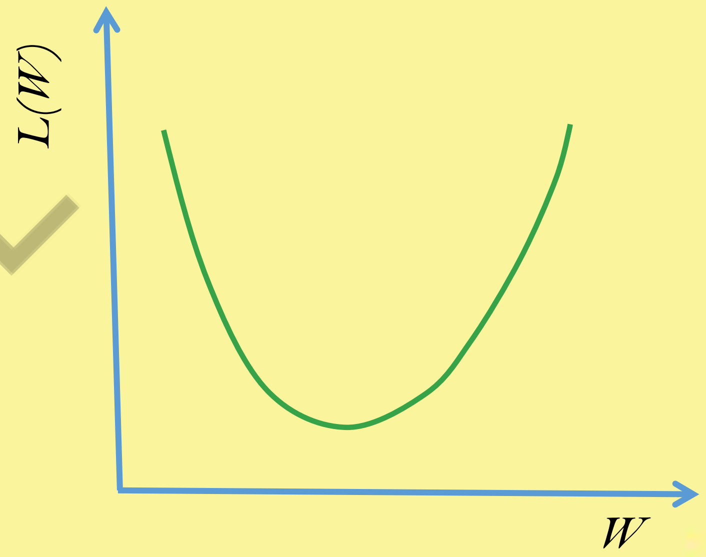
Challenges of Mini-batch Gradient Descent

- Choice of Proper Learning Rate:
  - Too small a learning rate leads to slow convergence.
  - A large learning rate may lead to oscillation around the minima or may even diverge.

# Gradient Descent Challenges

## Challenges of Mini-batch Gradient Descent

- Choice of Proper Learning Rate:
  - Too small a learning rate leads to slow convergence.
  - A large learning rate may lead to oscillation around the minima or may even diverge.

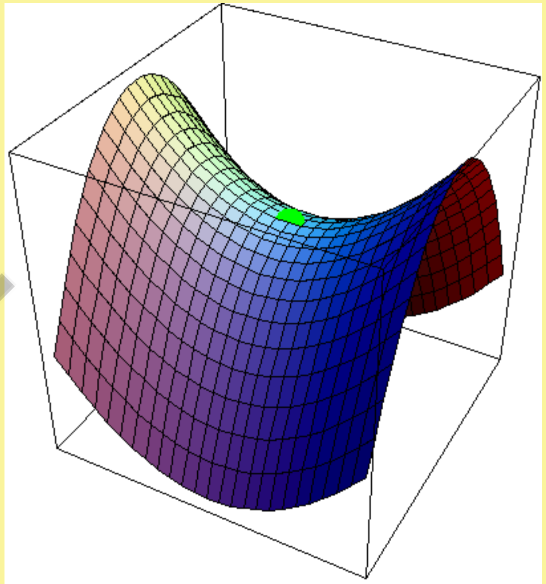


# Gradient Descent Challenges

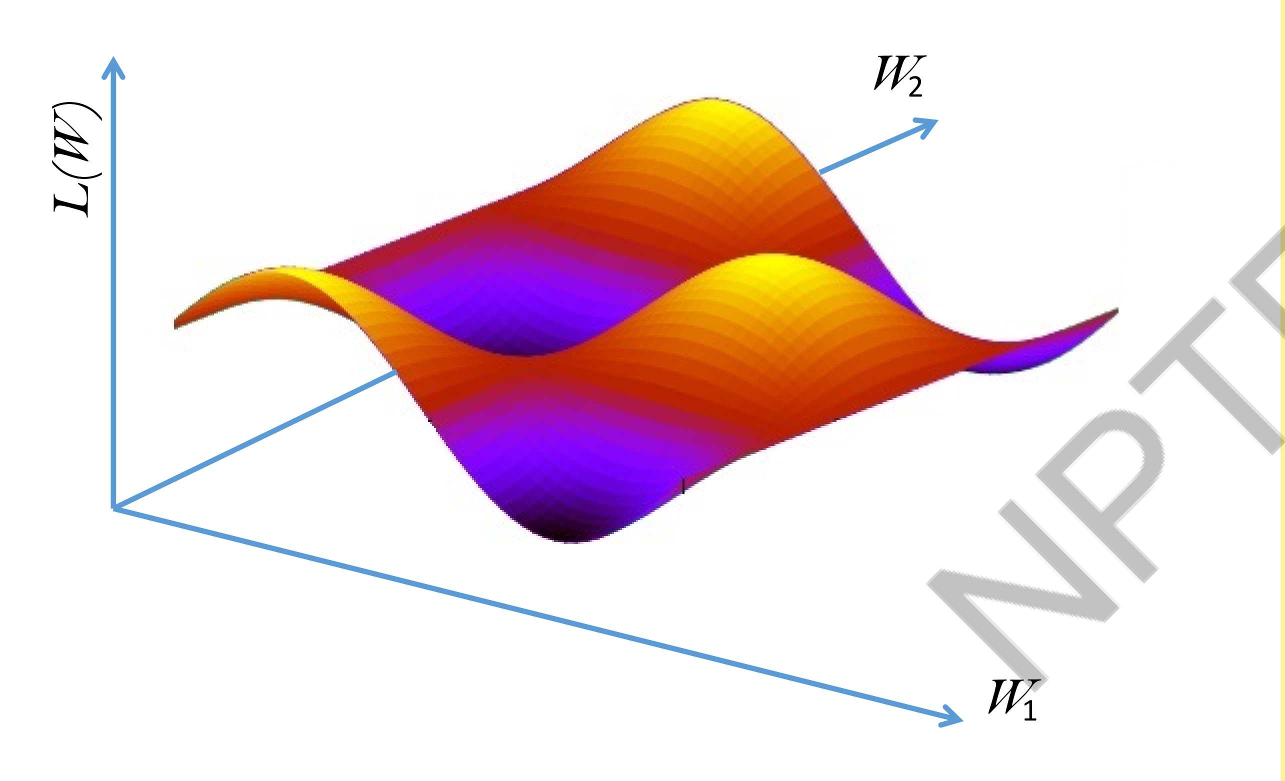
- Learning Rate Schedules: changing learning rate according to some predefined schedule.
  - The same learning rate applies to all parameter updates.
  - The data may be sparse and different features have very different frequencies.
  - Updating all of them to the same extent might not be proper.
  - Larger update for rarely occurring features might be a better choice.

# Gradient Descent Challenges

- Avoiding getting trapped in suboptimal local minima.
  - Difficulty arises in from saddle points, i.e. points where one dimension slopes up and another slopes down.
  - These saddle points are usually surrounded by a plateau of the same error, which makes it hard for SGD to escape, as the gradient is close to zero in all dimensions.



# Momentum Optimizer




# NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

# Topic

## Lecture 43: Popular Optimizing Gradient Descent

# Challenges

- ❑ Deep learning is data hungry.
- ❑ Overfitting or lack of generalization.
- ❑ Vanishing/Exploding Gradient Problem.
- ❑ Appropriate Learning Rate.
- ❑ Covariate Shift.
- ❑ Effective training.

CONCEPTS COVERED

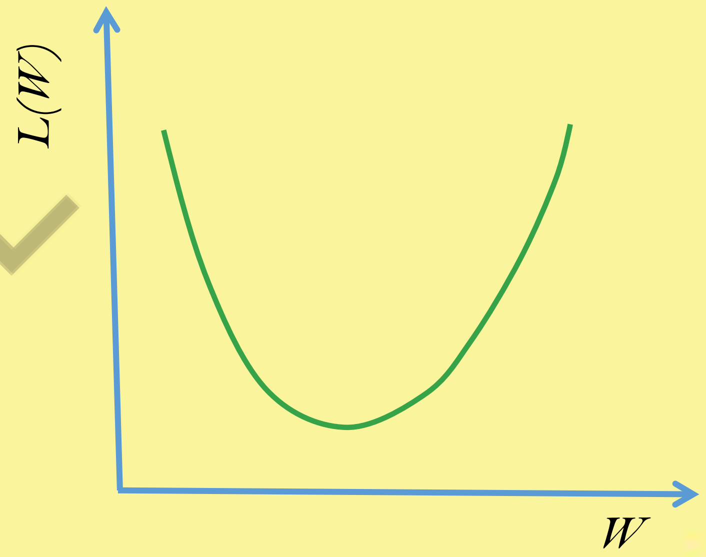
Concepts Covered:

- CNN
  - ResNet
  - Gradient Descent Challenges
  - Momentum Optimizer
  - Nestevor Accelerated Gradient
  - Adagrad.
  - etc.

# Gradient Descent Challenges

## Challenges of Mini-batch Gradient Descent

- ❑ Choice of Proper Learning Rate:
  - ❑ Too small a learning rate leads to slow convergence.
  - ❑ A large learning rate may lead to oscillation around the minima or may even diverge.

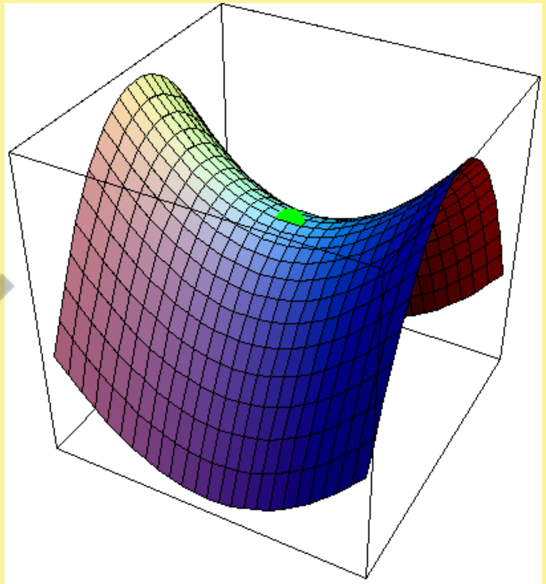


# Gradient Descent Challenges

- Learning Rate Schedules: changing learning rate according to some predefined schedule.
  - The same learning rate applies to all parameter updates.
  - The data may be sparse and different features have very different frequencies.
  - Updating all of them to the same extent might not be proper.
  - Larger update for rarely occurring features might be a better choice.

# Gradient Descent Challenges

- Avoiding getting trapped in suboptimal local minima.
  - Difficulty arises from saddle points, i.e. points where one dimension slopes up and another slopes down.
  - These saddle points are usually surrounded by a plateau of the same error, which makes it hard for SGD to escape, as the gradient is close to zero in all dimensions.



# Optimizing Gradient Descent

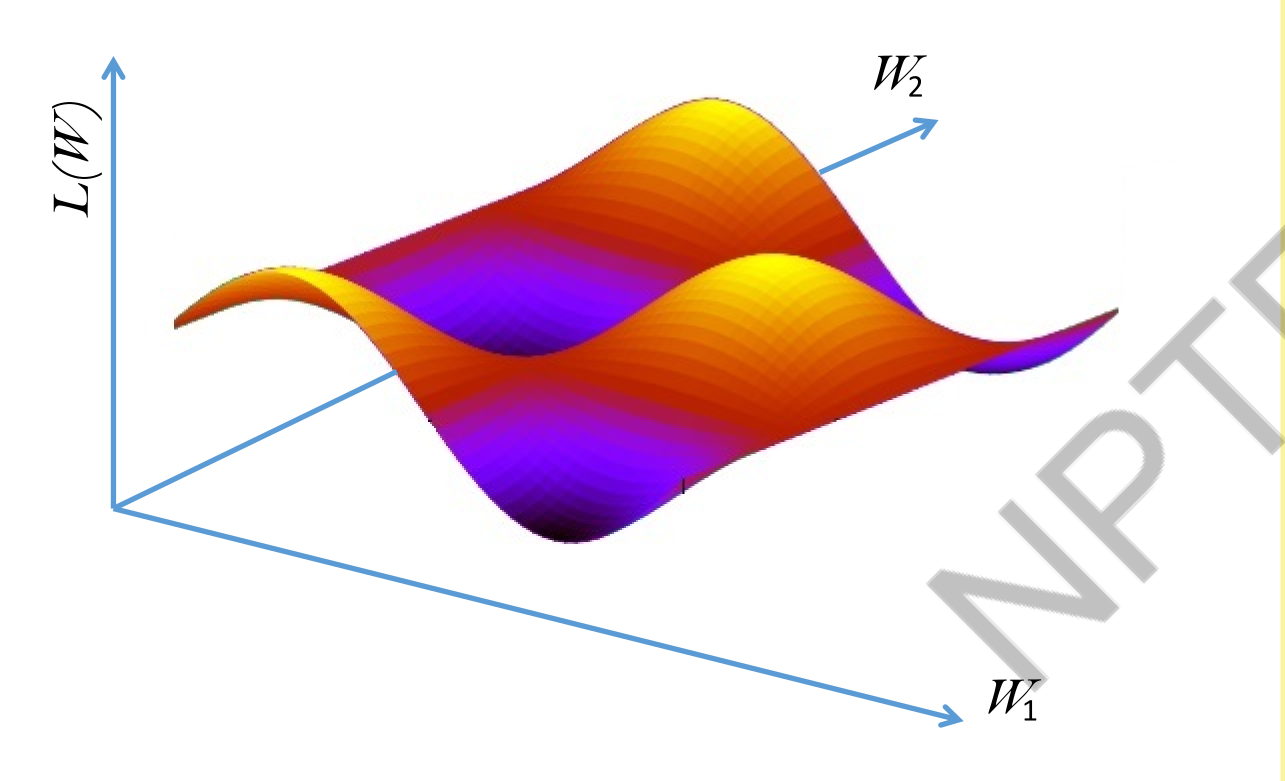
CONCEPTS COVERED

Concepts Covered:

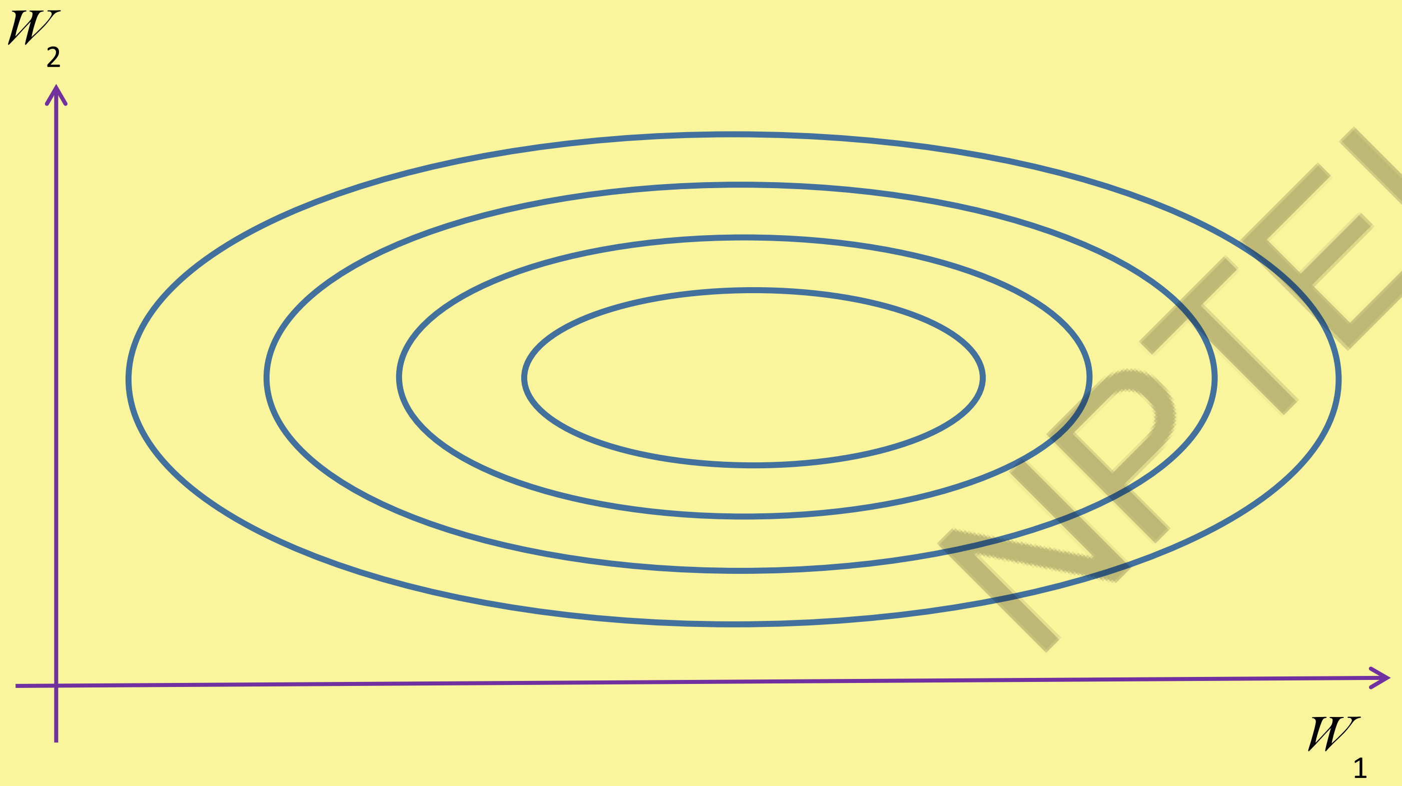
- CNN
  - ResNet
  - Gradient Descent Challenges
  - Momentum Optimizer
  - Adagrad.
  - etc.

# Momentum Optimizer

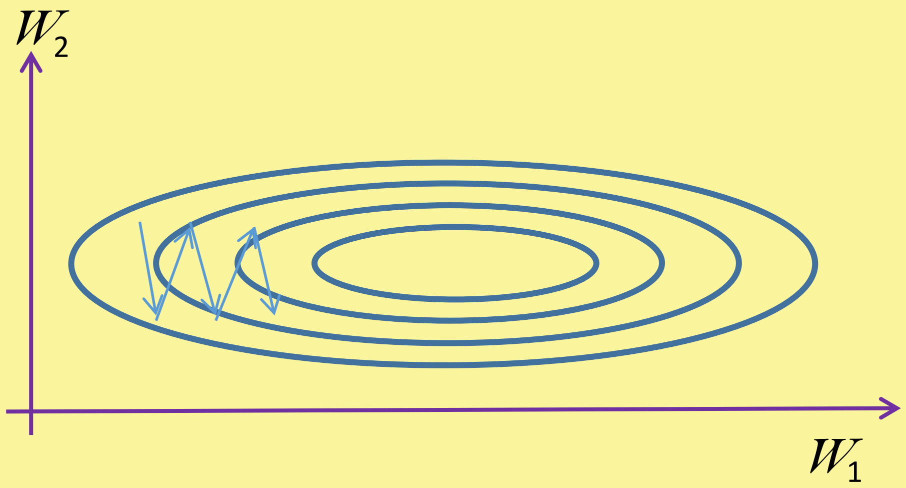
# Momentum Optimizer



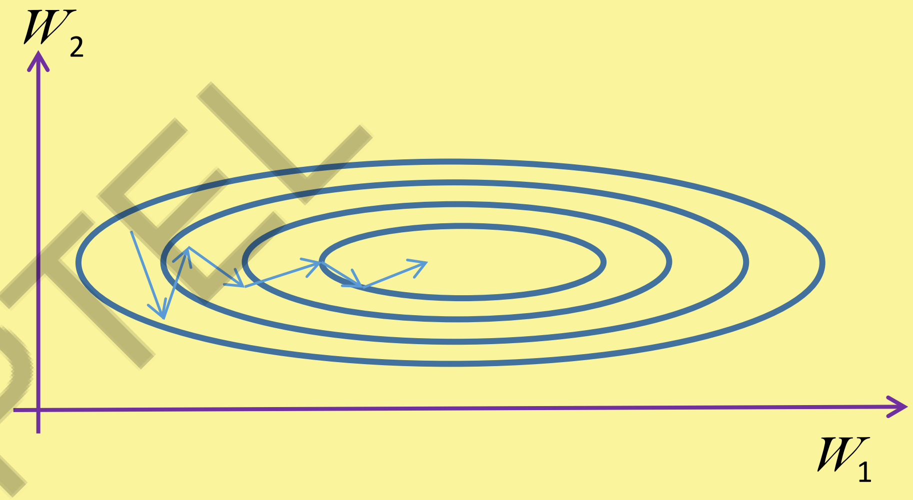
# Momentum Optimizer



# Momentum Optimizer



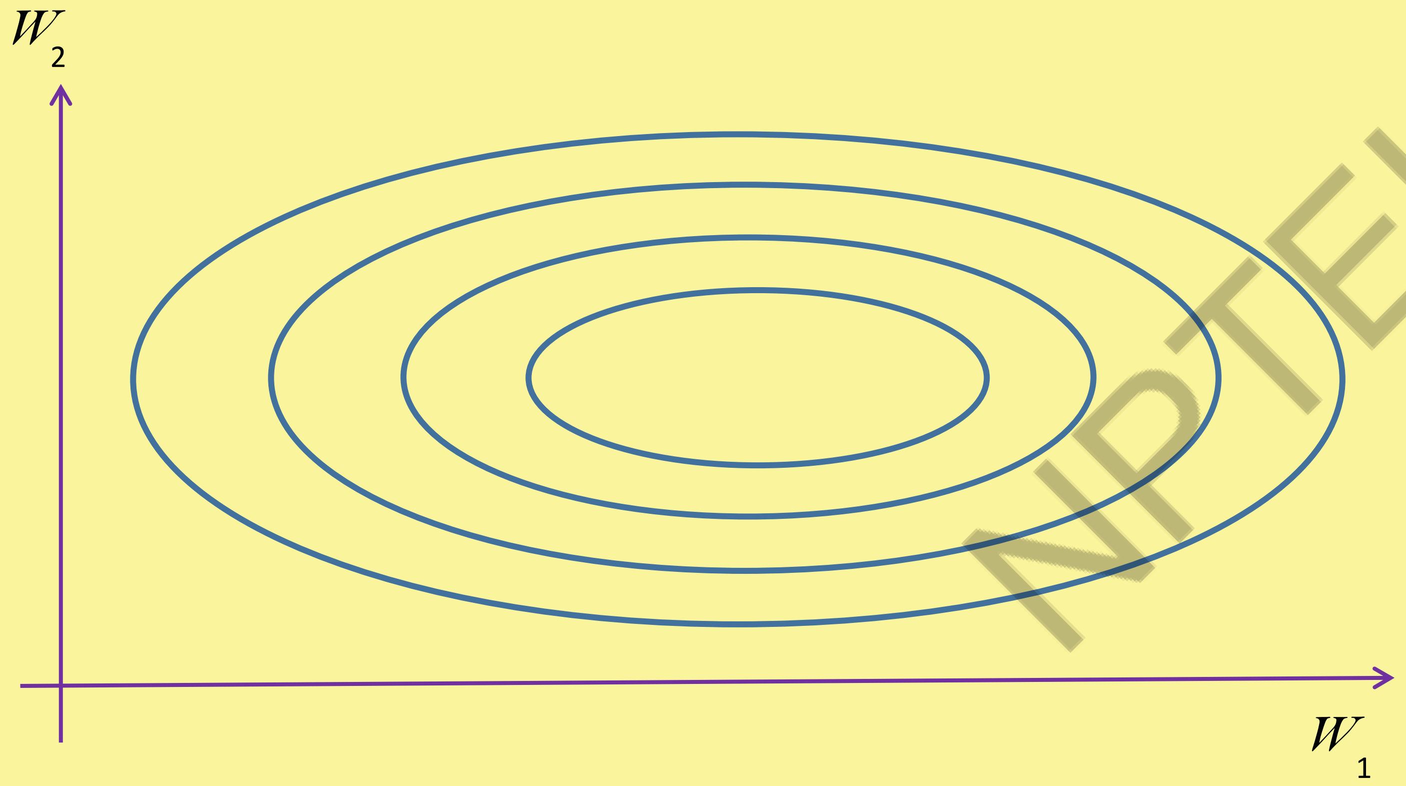

SGD




SGD with Momentum

# Nesterov Accelerated Gradient (NAG)

# Nesterov Accelerated Gradient (NAG)



# Problem with Momentum Optimizer/NAG

- Both the algorithms require the hyper-parameters to be set manually.
- These hyper-parameters decide the learning rate.
- The algorithm uses same learning rate for all dimensions.
- The high dimensional (mostly) non-nonconvex nature of loss function may lead to different sensitivity on different dimension.
- We may require learning rate could be small in some dimension and large in another dimension.

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur



## Topic

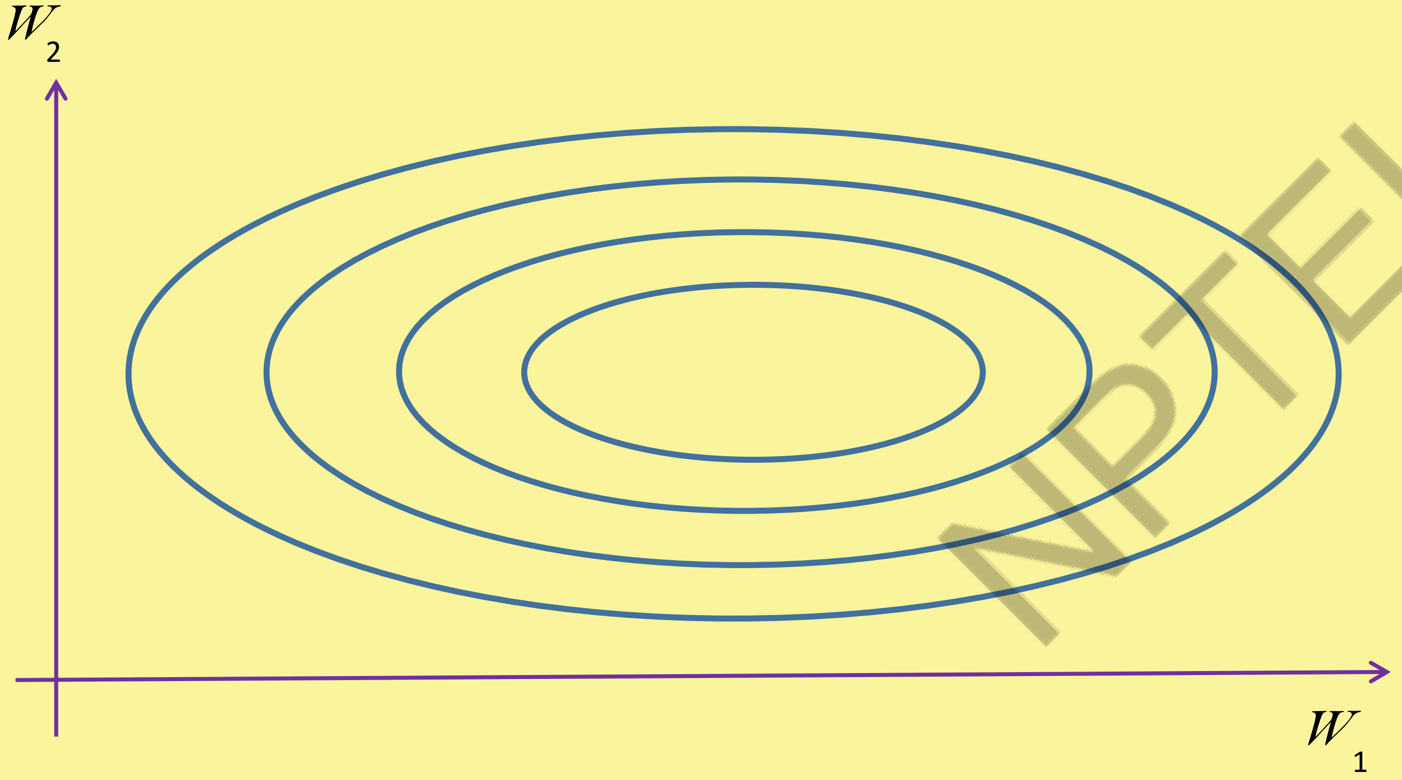
Lecture 44: Optimizing Gradient Descent II

# CONCEPTS COVERED

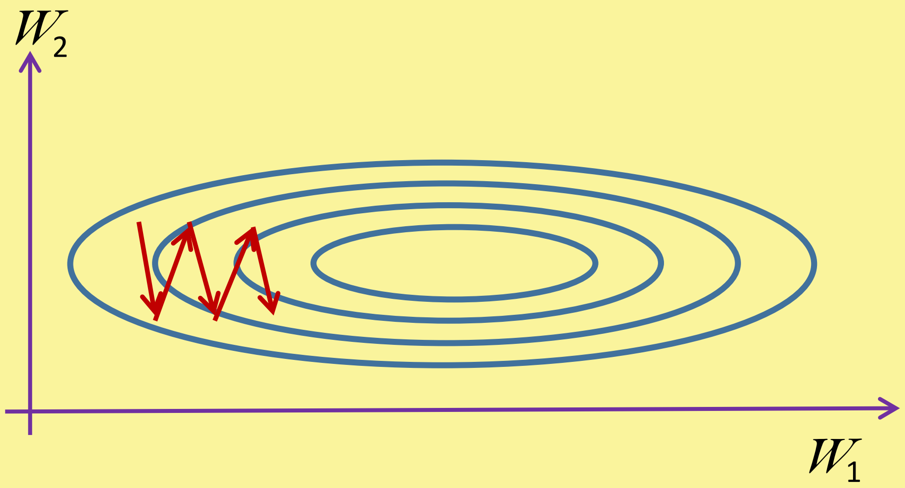
Concepts Covered:

- CNN
  - Gradient Descent Challenges
  - Momentum Optimizer
  - Nesterov Accelerated Gradient
  - Adagrad
  - RMSProp
  - etc.

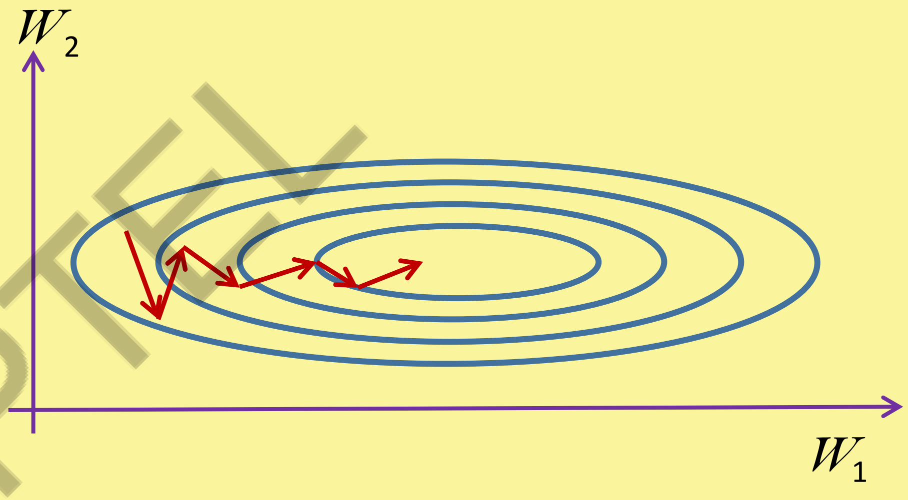
# Momentum Optimizer



# Momentum Optimizer



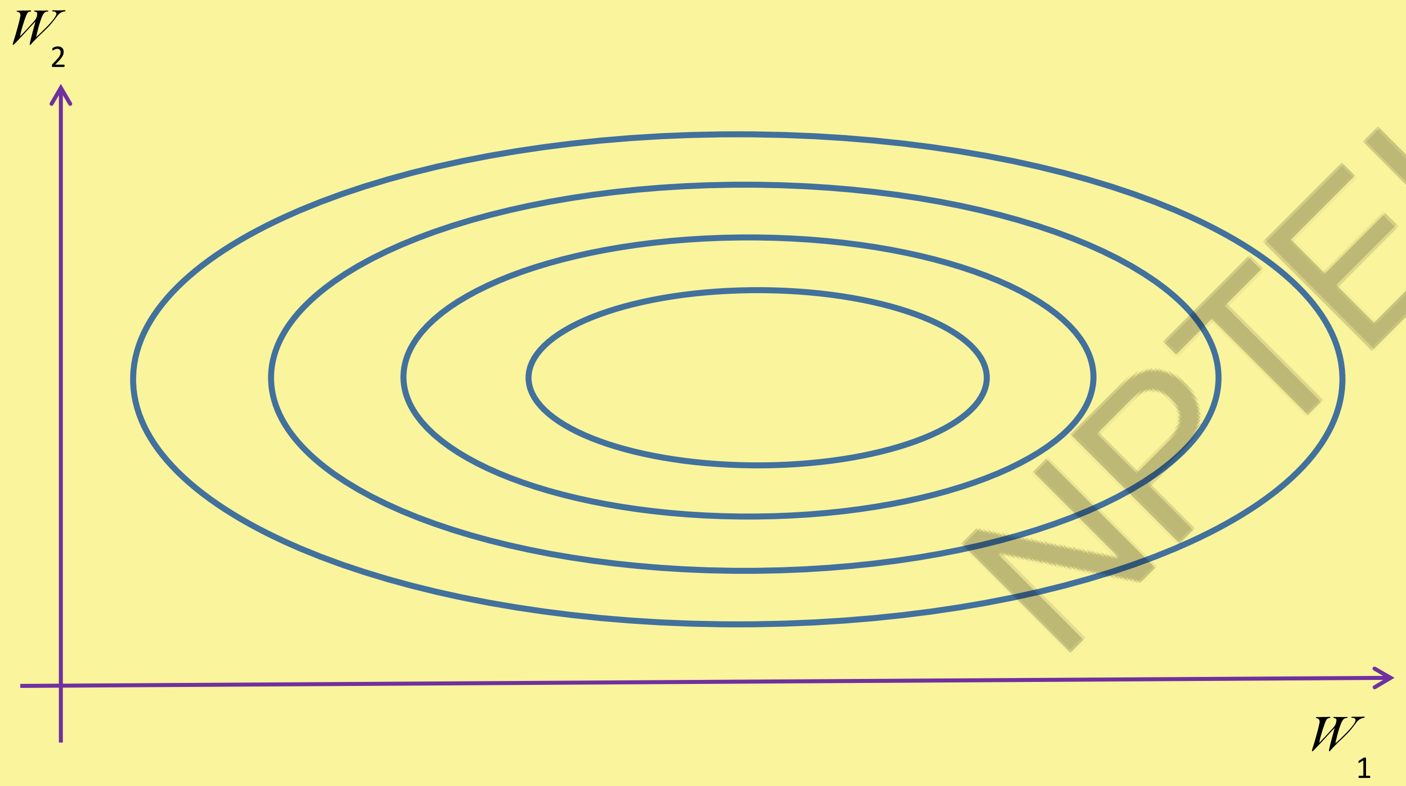

SGD




SGD with Momentum

# Nesterov Accelerated Gradient (NAG)

# Nesterov Accelerated Gradient (NAG)



# Problem with Momentum Optimizer/NAG

- Both the algorithms require the hyper-parameters to be set manually.
- These hyper-parameters decide the learning rate.
- The algorithm uses same learning rate for all dimensions.
- The high dimensional (mostly) non-nonconvex nature of loss function may lead to different sensitivity on different dimension.
- We may require learning rate be small in some dimension and large in another dimension.

# Adagrad

# Adagrad

- Adagrad adaptively scales the learning rate for different dimensions.
- Scale factor of a parameter is inversely proportional to the square root of sum of historical squared values of the gradient.
- The parameters with the largest partial derivative of the loss will have rapid decrease in their learning rate.
- Parameters with small partial derivatives will have relatively small decrease in learning rate.

# Adagrad

$$g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(W_t, X) \qquad r_t = \sum_{\tau=1}^{t} g_\tau \circ g_\tau$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\in I + r_t}} \circ g_t$$

$$\circ \rightarrow \text{element - wise product}$$

# Adagrad

$$\begin{bmatrix} W_{t+1}^{(1)} \\ W_{t+1}^{(2)} \\ \vdots \\ W_{t+1}^{(d)} \end{bmatrix} = \begin{bmatrix} W_{t}^{(1)} \\ W_{t}^{(2)} \\ \vdots \\ W_{t}^{(d)} \end{bmatrix} - \begin{bmatrix} \dfrac{\eta}{\sqrt{\in + r_t^{(1)}}} . g_t^{(1)} \\ \dfrac{\eta}{\sqrt{\in + r_t^{(2)}}} . g_t^{(2)} \\ \vdots \\ \dfrac{\eta}{\sqrt{\in + r_t^{(d)}}} . g_t^{(d)} \end{bmatrix}$$

# Adagrad

Positive Side:

- Adagrad adaptively scales the learning rate for different dimensions by normalizing with respect to the gradient magnitude in the corresponding dimension.
- Adagrad eliminates the need to manually tune the learning rate.
- Reduces learning rate faster for parameters showing large slope and slower for parameters giving smaller slope.
- Adagrad converges rapidly when applied to convex functions.

# Adagrad

Negative side:

- If the function is non-convex:- trajectory may pass through many complex terrains eventually arriving at a locally region.
- By then learning rate may become too small due to the accumulation of gradients from the beginning of training.
- So at some point the model may stop learning.


NPTEL ONLINE CERTIFICATION COURSES

Thank you

# NPTEL ONLINE CERTIFICATION COURSES

## Course Name: Deep Learning


Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur


## Topic

**Lecture 45: Optimizing Gradient Descent III**

# CONCEPTS COVERED

Concepts Covered:

- CNN
  - Gradient Descent Challenges
  - Momentum Optimizer
  - Nesterov Accelerated Gradient
  - Adagrad
  - RMSProp
  - etc.

# Adagrad

$$g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(W_t, X) \qquad r_t = \sum_{\tau=1}^{t} g_\tau \circ g_\tau$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\in I + r_t}} \circ g_t$$

$\circ \rightarrow$ element - wise product

# Adagrad

Positive Side:

- Adagrad adaptively scales the learning rate for different dimensions by normalizing with respect to the gradient magnitude in the corresponding dimension.
- Adagrad eliminates the need to manually tune the learning rate.
- Reduces learning rate faster for parameters showing large slope and slower for parameters giving smaller slope.
- Adagrad converges rapidly when applied to convex functions.

# Adagrad

Negative side:

- If the function is non-convex:- trajectory may pass through many complex terrains eventually arriving at a locally region.
- By then learning rate may become too small due to the accumulation of gradients from the beginning of training.
- So at some point the model may stop learning.

# RMSProp

# RMSProp

- RMSProp uses exponentially decaying average of squared gradient and discards history from the extreme past.
- Converges rapidly once it finds a locally convex bowl.
- Treats this as an instance of Adagrad algorithm initialized within that bowl.

# RMSProp

$$g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(W_t, X)$$

$$r_t = \beta r_{t-1} + (1-\beta) g_t \circ g_t$$ → Exponentially decaying average

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\in I + r_t}} \circ g_t$$

# RMSProp with Nesterov Momentum

$$\tilde{W} = W_t + \alpha v \qquad\qquad g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(\tilde{W}, X)$$

$$r_t = \beta r_{t-1} + (1-\beta) g_t \circ g_t$$

$$v_{t+1} = \alpha v_t - \frac{\eta}{\sqrt{\in I + r_t}} \circ g_t \qquad\qquad W_{t+1} = W_t + v_t$$

# Adaptive Moments (Adam)

# Adam

- Variant of the combination of RMSProp and Momentum.
- Incorporates first order moment (with exponential weighting) of the gradient (Momentum term).
- Momentum is incorporated in RMSProp by adding momentum to the rescaled gradients.
- Both first and second moments are corrected for bias to account for heir initialization to zero.

# Adam

$$g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(W, X)$$

Biased first and second moments

$$s_t = \beta_1 s_{t-1} + (1 - \beta_1) g_t$$

$$r_t = \beta_2 r_{t-1} + (1 - \beta_2) g_t \circ g_t$$

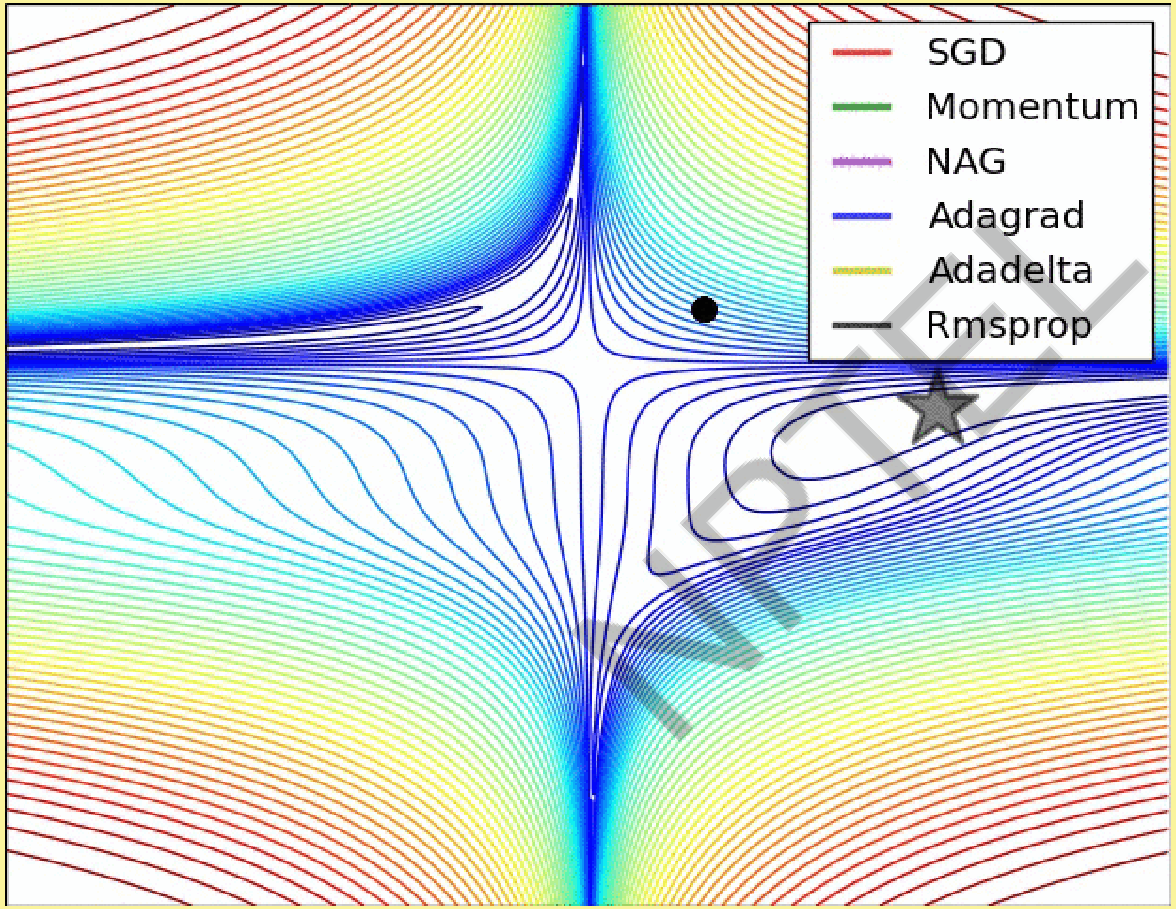
# Adam

Bias corrected first and second moments

$$\hat{s}_t = \frac{s_t}{1-\beta_1} \qquad \hat{r}_t = \frac{r_t}{1-\beta_2}$$

$$W_{t+1} = W_t - \eta \frac{\hat{s}_t}{\sqrt{\in I + \hat{r}_t}}$$

# Momentum Optimizer




Animation Source:-
https://imgur.com/a/Hqolp


# NPTEL ONLINE CERTIFICATION COURSES

Thank you