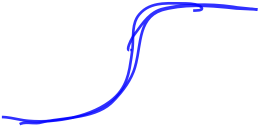
SMAI-M20-L31: Learning in Neural Networks

C. V. Jawahar

IIIT Hyderabad

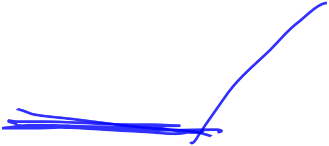
October 28, 2020

Class Review



1. ReLu activation in Neural Networks; Its properties, its derivative, its limitations, etc.

$\max(0, x)$



Blank

Recap:

- **Supervised Learning:** Formulation, Conceptual Issues, Concerns etc. (i) Loss Functions and Optimization (ii) Probabilistic View, Bayesian View, MLE (iii) Eigen Vector based optimization (iv) Gradient Descent: Stochastic and Batch GD (v) Classification and Regression
- **Classifiers:** (i) Nearest Neighbour, (ii) Notion of a Linear Classifier (iii) Perceptrons (iv) Bayesian Optimal Classifier (v) Logistic Regression (vi) Multiclass classification architectures (v) Decision Trees (vi) SVMs (hard margin, soft margin, kernel) (vii) Kernel trick and kernelized algorithms
- **Dimensionality Reduction and Applications:** (i) Feature Selection and Extraction (ii) PCA (iii) LDA (iv) Eigen face
- **Matrix Factorization and Applications:** (i) SVD, (ii) Eigen Decomposition (iii) Matrix Completion (iv) LSI (v) Recommendations
- **Neural Network Architectures and Learning** (i) Neuron model, Single Layer Perceptrons (ii) SLP(iii) MLP (iv) Backpropagation (v) Chain rule (vi) Activations (vii) challenges in optimization

This Lecture:

FC ⟷ Conv

1. **Convolution Layer**
   1. A very popular layer, specially for image, speech and text data. (where local context is very important)
   2. Connection to signal processing, filtering.
   3. Do read[1]
   4. (out of scope)But strongly advised to read[2]
2. **Momentum**
   1. Improvement over gradient descent updates
   2. May read later [3]

para

$10^8 \longrightarrow 3$

Questions? Comments?

$w^{k+1} \leftarrow w^k - \eta \nabla J$

F.C

m n

[1] https://danieltakeshi.github.io/2019/03/09/conv-matmul/
[2] https://cs231n.github.io/convolutional-networks/
[3] http://d2l.ai/chapter_optimization/momentum.html

Blank

Blank

Blank

Discussions Point - I

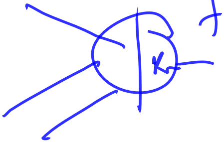
0 1 2 −3 2 . . . .

3 , 0 , ...

Consider an input of $1, 2, -3, 2, -1, 0, -4, 6$. Assume zero padding.

1. Assuming the layer as convolution-layer with weights as $1, 1, 1$, find the convolution output.
2. Consider two convolution weights (i) $1, 1, 1$ and (ii) $1, 0, -1$. All the neurons have ReLU activation. Find the outputs.
3. Consider a stride of two [4]. Find the outputs?

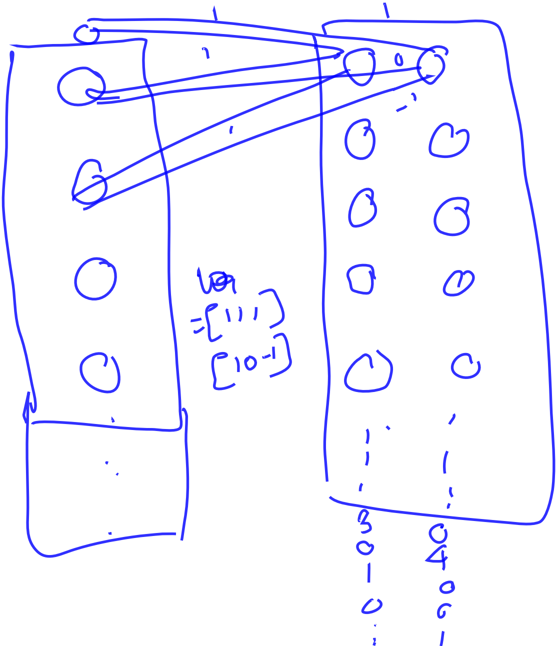
$\max(0, x)$

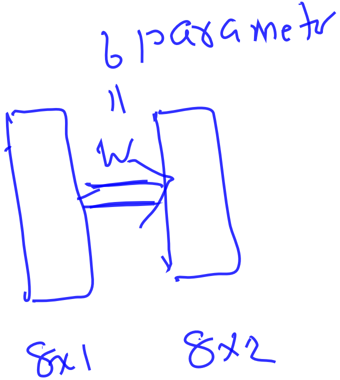


$f\left(\sum_i w_i x_i\right)$

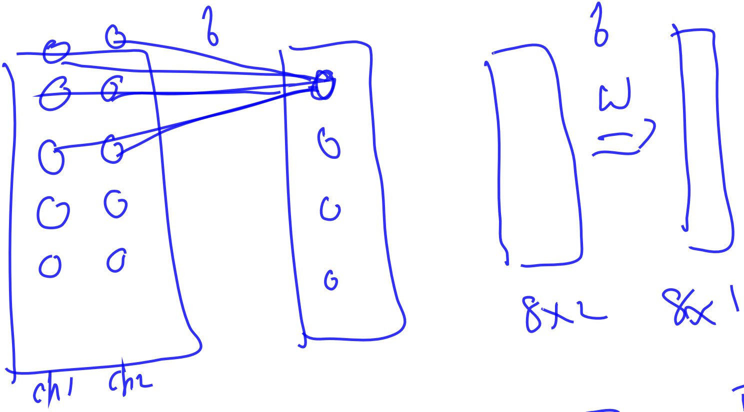
[4] When the stride is 1 (that is what we did till now), then we move the filters by 1 sample at a time. When the stride is 2 then we move the filters by 2 samples. Effectively output is half the size of the input.

Blank

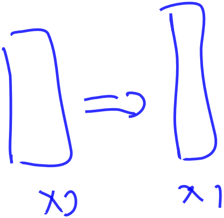


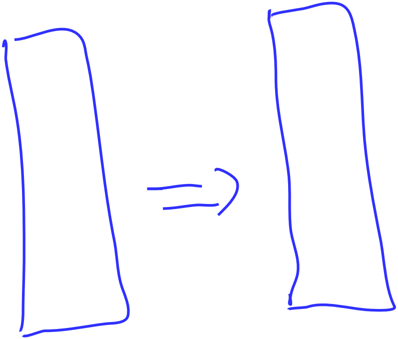


Blank



RGB Colour image





$n \times p \Rightarrow n \times q$

m

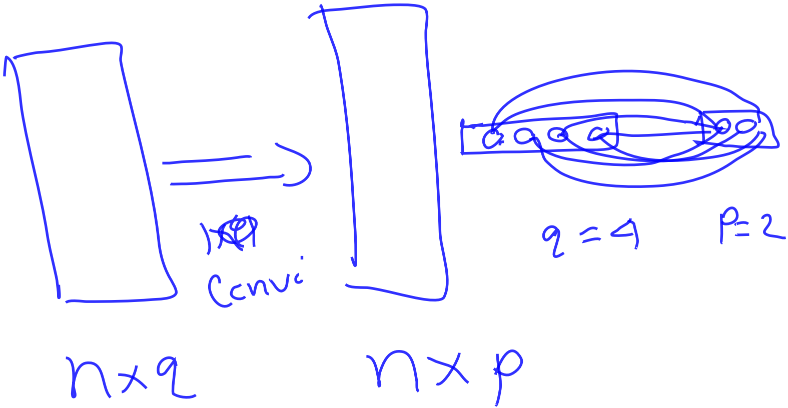
Conv size

#par

stride

$m \ll n$

what is 1x1 Conv?



Discussions Point -II

We know the learning rule as:

$$w^{k+1} \leftarrow w^k - v_k$$

$$v_k = \eta \nabla J + \beta v_{k-1}$$

1. What if $\eta = 1.0$ and $\beta = 0.0$?
2. What if $\eta = 0.0$ and $\beta = 1.0$?
3. Assume $\nabla J = -0.1$ for all $k = 0, 1, \ldots, 10$. $v_{-1} = 0$. $w^0 = 0.0$. $\eta = 0.1$ and $\beta = 0.9$, what happens to w^k for $k = 1, 2, 3$?
4. Where it should have reached with and without momentum for $k = 10$? (Appreciate how momentum helps in speeding up, if we have a consistent slow slope).

Blank

Blank

What Next:?

1. NN Architectures and NN Learning
2. Programming for Deep Learning.