

VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE
IV/IV B.Tech Degree Examination, March,2021
Seventh Semester
INFORMATION TECHNOLOGY
17IT4703A DEEP LERNING

PART-A

10X1=10M

1.a. List the types of layers in CNN.

Input, convolution, pooling, fully connected, softmax (output)

b. What is the difference between deep and shallow network?

Shallow neural networks is a NN that usually have only one hidden layer where as a deep NN have several hidden layers

c. what is word embedding?

Word embedding is representing a word in the form of a vector or encoding a word

d. How to represent filters and feature maps as neurons in a convolution layer?

Layers of neurons in a feedforward neural net represent either the original image or a feature map. Filters represent combinations of connections that get replicated across the entirety of the input. A neuron in the feature map is activated if the filter contributing to its activity detected an appropriate feature at the corresponding position in the previous layer.

e. Write the function of denoising autoencoder?

Denoising autoencoder generates embeddings that are resistant to noise.

f. What is sparsity in autoencoder?

A sparse autoencoder is simply an autoencoder whose training criterion involves a sparsity penalty to improve interpretability.

g. Define computational graph

A computational graph is an unfolding of a recursive or recurrent neural network that has a repetitive structure

h. Write short notes on bidirectional RNNs

Bidirectional RNNs combine an RNN that moves forward through time beginning from the start of the sequence with another RNN that moves backward through time beginning from the end of the sequence.

i. What is overfitting in deep neural networks?

It refers to the phenomenon where a neural network models the training data very well but fails when it sees new data from the same problem domain

J. Write the difference between feed forward network and recurrent network.

Feedforward neural networks pass the data forward from input to output, while recurrent networks have a feedback loop where data can be fed back into the input at some point before it is fed forward again for further processing and final output.

[Convolutional Neural Network. Learn Convolutional Neural Network from... | by dshahid380 | Towards Data Science](#)

[BLOG - DebuggerCafe](#)

PART-B

4X15=60M

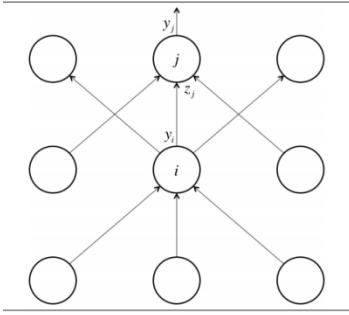
UNIT-I

2. a. Derive the back propagation rule considering the training rule for output unit weights and training rule for hidden unit weights.

8M

Description of feed forward and back propagation concept- 2M

Derivation-6M



If we calculate the error function derivatives at the output layer:

$$E = \frac{1}{2} \sum_{j \in \text{output}} (t_j - y_j)^2 \Rightarrow \frac{\partial E}{\partial y_i} = -(t_j - y_j)$$

partial derivative of the logit with respect to the incoming output data from the layer beneath is merely the weight of the connection w_{ij} ,

$$\frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial z_j} \frac{dz_j}{dy_i} = \sum_j w_{ij} \frac{\partial E}{\partial z_j}$$

$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \frac{dy_j}{dz_j} = y_j(1 - y_j) \frac{\partial E}{\partial y_j}$$

Combining these two together, the error derivatives of layer i in terms of the error derivatives of layer j:

$$\frac{\partial E}{\partial y_i} = \sum_j w_{ij} y_j(1 - y_j) \frac{\partial E}{\partial y_j}$$

to modify the weights after each training example:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial z_j}{\partial w_{ij}} \frac{\partial E}{\partial z_j} = y_i y_j(1 - y_j) \frac{\partial E}{\partial y_j}$$

by summing up the partial derivatives over all the training examples in our dataset. This gives us the following modification formula to update weights .

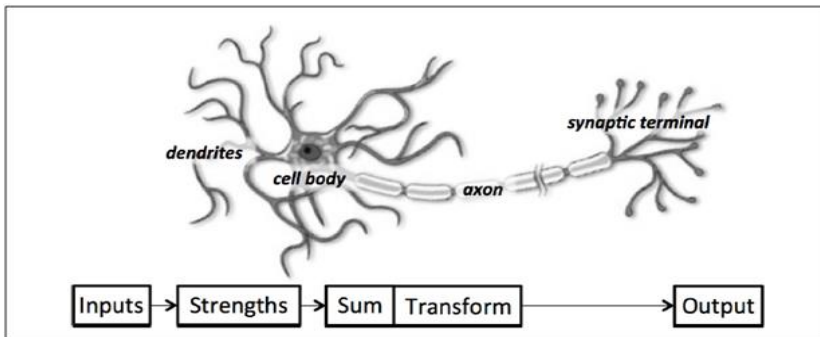
$$\Delta w_{ij} = - \sum_{k \in \text{dataset}} \epsilon_i^{(k)} y_j^{(k)} (1 - y_j^{(k)}) \frac{\partial E^{(k)}}{\partial y_j^{(k)}}$$

b. Write the functional description of a biological neuron's structure with a suitable diagram.

7M

Diagram : 3M

Description : 4M



(OR)

3. a. What is gradient descent ? Explain the following three variants of gradient descent. 9M

(i) Batch

(ii) Stochastic

(iii) Mini-batch

Gradient descent—1.5M

Explanation of each type —3x2.5=7.5M

In gradient descent the error surface is computed and then it follows the gradient to take the path of steepest descent. Using the Gradient Descent optimization algorithm, the weights are updated incrementally after each epoch.

In batch gradient descent, the entire dataset is used to compute the error surface.

In Stochastic Gradient Descent we update the weights after each training sample.

Mini-batch gradient descent performs an update for every mini-batch of n training examples.

b. List and Explain various activation functions. 6M

List- 1M

Explanation with formulas- 4M

Output shapes- 1M

Step

Sigmoid

Relu

Tanh

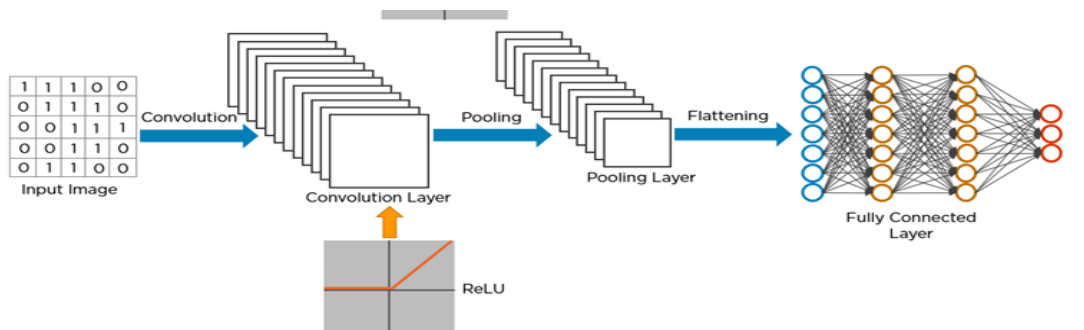
UNIT-II

4. a. Draw and explain the architecture of convolution neural network.

7M

Diagram -- 3M

Description of layers- 4M



b. How batch normalization can be expressed for a convolutional layer? Express in TensorFlow.

8M

Steps explanation -4M

Explanation of Tensor flow Function-- 4M

Batch normalization takes this a step further by normalizing inputs to every layer in our neural network. we modify the architecture of network to include operations that:

1. Grab the vector of logits incoming to a layer before they pass through the nonlinearity
2. Normalize each component of the vector of logits across all examples of the mini- batch by subtracting the mean and dividing by the standard deviation (we keep track of the moments using an exponentially weighted moving average)
3. Given normalized inputs \hat{x} , use an affine transform to restore representational power with two vectors of (trainable) parameters: $\gamma\hat{x} + \beta$

Expressed in TensorFlow, batch normalization can be expressed as follows for a convolutional layer:

```
def conv_batch_norm(x, n_out, phase_train)
```

(or)

5.a. Discuss about short comings of feature selection.

8M

Problems of deep neural nets for image classification—4M

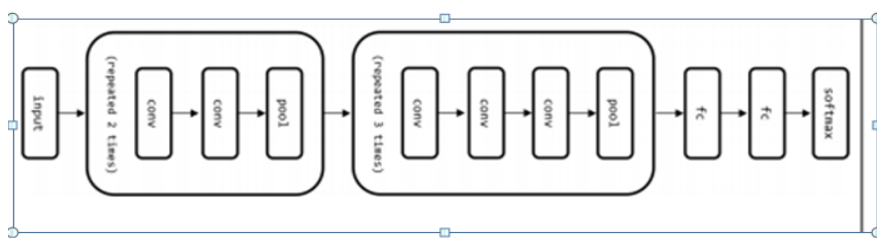
Explanation -4M

- In deep neural networks, each layer of a neural network is responsible for learning and building up features to represent the input data that it receives.
- If we attempt to tackle the image classification problem in this way, we will face a challenge of scaling well with deep neural networks .
- As images grow larger, input layer would more number of weights.
- And this needs to have lots of neurons over multiple layers, so these parameters add up quite quickly.
- The full connectivity is not only wasteful, but also more likely to overfit to the training dataset.

b. Write the architectural description of VGGNet, a deep convolutional network built for ImageNet. 7M

diagram- 3M

explanation -4M



The input to VGG based convNet is a 224*224 RGB image. Preprocessing layer takes the RGB image with pixel values in the range of 0–255 and subtracts the mean image values which is calculated over the entire ImageNet training set.

UNIT-III

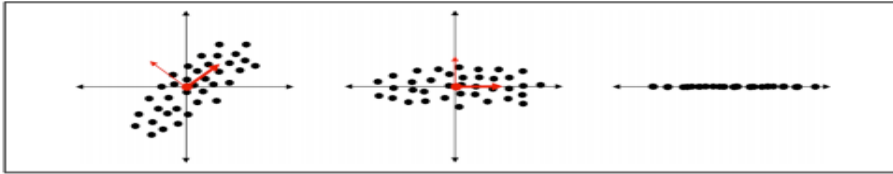
6. a. What is Principle Component Analysis algorithm? Write the applications of it.

8M

Expalnation of PCA-6M

Applications-2M

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.



Applications:

PCA is predominantly used as a dimensionality reduction technique in domains like facial recognition, computer vision and image compression.

It is also used for finding patterns in data of high dimension in the field of finance, data mining, bioinformatics, psychology, etc.

b. Why would we use autoencoders? Differentiate between autoencoders Vs. PCA.
7M

Use of auto encoder-2M

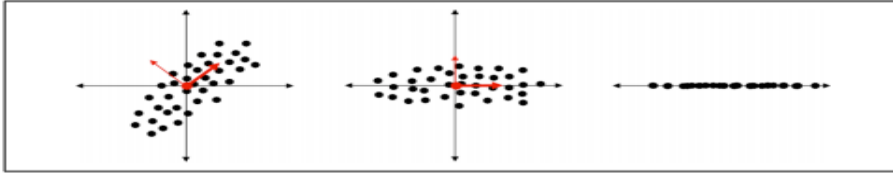
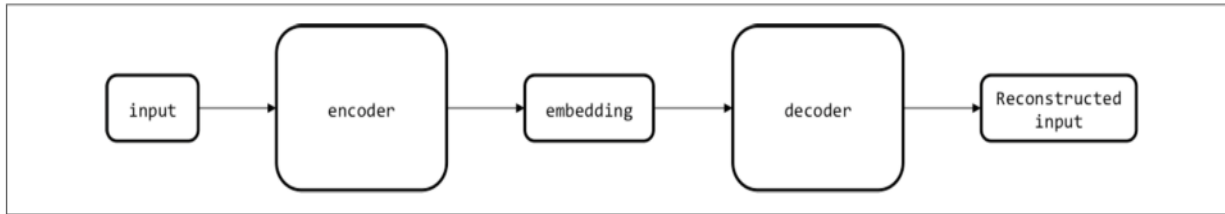
Differnces with brief explanation -5M

To geneate low-dimensional representations in unsupervised fashion, we use a new network architecture called auto encoder

PCA is essentially a linear transformation but Auto-encoders are capable of modelling complex non linear functions.

PCA is faster and computationally cheaper than autoencoders.

A single layered autoencoder with a linear activation function is very similar to PCA



(or)

7. a. what is word2vec and what it does? How the activation layer is computed in the word embedding?

8M

Explanation of word tovec- CBOW , skipgram , one hot encoding - 3M

NCE strategy -5M

The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text.

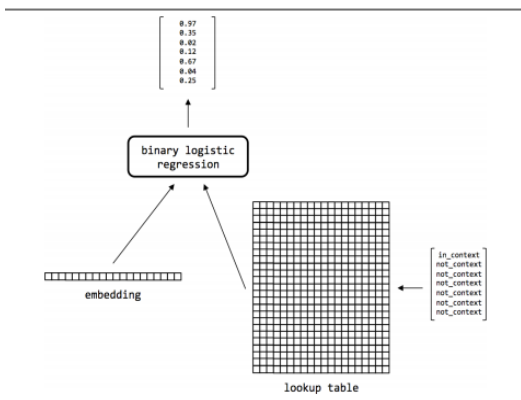
The first flavor of Word2Vec was the Continuous Bag of Words (CBOW) model. The CBOW model used the encoder to create an embedding from the full context (treated as one input) and predict the target word.

The second flavor of Word2Vec is the Skip-Gram model. The Skip-Gram model does the inverse of CBOW, taking the target word as an input, and then attempting to predict one of the words in the context.

The structure of the encoder is surprisingly simple. It is essentially a lookup table with V rows, where the i th row is the embedding corresponding to the i th vocabulary word.

The decoder is slightly trickier because we make some modifications for performance. The naive way to construct the decoder would be to attempt to reconstruct the one-hot encoding vector for the output, which we could implement with a run-of-the-mill feed-forward layer coupled with a softmax.

To reduce the number of parameters, decoder known as noise-contrastive estimation (NCE) is implemented



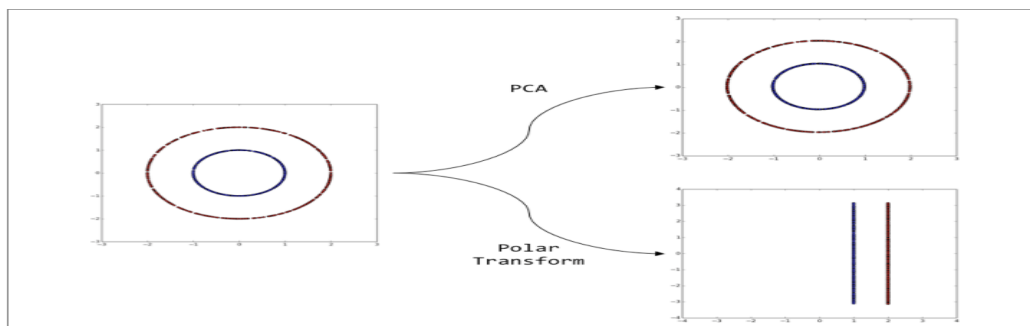
b. Elaborate a simulation in which PCA fails to optimally transform the data for dimensionality reduction? 7M

Any case study description -3M

Why it is difficult for PCA—4M

(Any non linear or complex example where PCA fails can be given marks)

PCA has been used for decades for dimensionality reduction, but it fails to capture important relationships that are piecewise linear or nonlinear.



The example shows data points selected at random from two concentric circles. There is no linear direction that contains more information here than another. Here Information is being encoded in a nonlinear way. PCA fails in capturing important relationships in complex datasets like this.

UNIT-IV

8.a . Why do you consider recurrent neural network? Also, explain what problems are normal CNNs good at?

8M

Why Rnn with brief explanation -5M

Explanation of problems normal CNNs good at -3M

- Recurrent neural networks or RNNs are a family of neural networks for processing sequential data.
- a neural network that is specialized for processing a sequence of values $x(1), \dots, x(\tau)$
- Process longer sequences of variable length.
- Each member of the
- output is a function of the previous members of the output

CNN

- CNN is a neural network that is specialized for processing a grid of values X such as an image
- Process images with large width and height
- output will not depend on previous members of the output

b. Explain the basic schema of an echo state network with a tunable frequency generator task.
7M

what is ESN—2M

Schema— 5M

Echo state network is a type of Recurrent Neural Network, part of the *reservoir computing* framework, which has the following particularities:

The weights between the input -the hidden layer (the 'reservoir') : W_{in} and also the weights of the 'reservoir': W_r are randomly assigned and not trainable.

The weights of the output neurons (the 'readout' layer) are trainable and can be learned so that the network can reproduce specific temporal patterns.

The hidden layer (or the 'reservoir') is very sparsely connected (typically $< 10\%$ connectivity) the reservoir architecture creates a recurrent **non linear** embedding (H on the image below) of the input which can be then connected to the desired output and these final weights will be trainable.

(or)

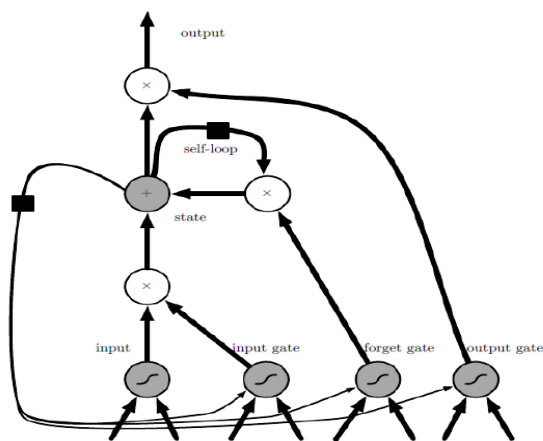
9.a . Explain gated recurrent neural networks. How are gated RNNs better than RNNs?

8M

Any gated rnn (LSTM) explanation -6M

How gated rnn s better than rnn -2M

In gated RNN there are generally three gates namely Input/Write gate, Keep/Memory gate and Output/Read gate and hence the name gated RNN for the algorithm. These gates are responsible for allowing/disallowing the flow of signal from the respective states.



Gated RNN s solve the problem of vanishing or exploding gradients.

Gated RNNs are based on the idea of creating paths through time that have derivatives that neither vanish nor explode.

b. what is a computational graph? Explain a more complex unfolded computational graph with diagram.

7M

Computational graph- 1M

Any RNN model and its unfolded diagram with explanation – 6M

A computational graph is an unfolding of a recursive or recurrent neural network that has a repetitive structure.

