Deep Learning Course – L 7 - Detailed Notes

Fully Connected Neural Networks (FCNN), Forward & Backward Propagation

---

🧠 1. Recap of Key Concepts

Before diving into fully connected neural networks, it's important to understand some foundational concepts:

- **Logistic Regression**: A simple neural network used for binary classification.
- **Cost Function**: Measures the difference between predicted and true values.
- **Gradient Descent**: Optimizes the parameters (weights, biases) to reduce the cost.
- **Forward Propagation**: Flow of inputs through the network to generate a prediction.
- **Backward Propagation**: Flow of gradients back through the network to update weights.

🔁 2. Forward Propagation in FCNN

🎯 Objective:

To compute the output of the neural network and calculate the **loss** (error).

Steps in Forward Propagation:

✅ *Step 1: Linear Transformation (Calculate z)*
z=W·X+Bz = W \cdot X + Bz=W·X+B

- WWW: Weight matrix
- XXX: Input vector
- BBB: Bias vector

Each layer applies this operation to its inputs.

✅ *Step 2: Activation Function*

a=g(z)a = g(z)a=g(z)

- ggg: Non-linear activation function (ReLU, Sigmoid, etc.)
- aaa: Activation (output of the current layer)

□ Weight Matrix Dimensions:

If:

- Inputs = 4 features
- Hidden units = 3 neurons
  Then:

W:3×4(each neuron has 4 weights)W: 3 \times 4 \quad \text{(each neuron has 4 weights)}W:3×4(each neuron has 4 weights)

Another example:

- Layer 2 has 2 hidden units
- Inputs to this layer = 3
  Then:

W[2]:2×3W^{[2]}: 2 \times 3W[2]:2×3

🎯 Final Output:

- The output layer gives us y^\hat{y}y^ (predicted value).
- We calculate **loss** by comparing y^\hat{y}y^ with actual label yyy.

📉 Loss Functions (based on problem type):

- **Binary Classification**: Negative Log Loss (Binary Cross-Entropy)
- **Regression**: Mean Squared Error (MSE)
- **Multi-class Classification**: Categorical Cross-Entropy

🔄 3. Backward Propagation in FCNN

🎯 Objective:

To update the weights using **gradients** of the loss with respect to the weights and biases.

Steps in Backward Propagation:

1. **Compute Derivatives**:
    - Derivatives of the loss w.r.t weights and biases (using chain rule).
2. **Apply Gradient Descent**:

    $$W := W - \alpha \cdot \frac{\partial \text{Loss}}{\partial W}$$

    - α: Learning rate
    - Update is done in all layers from output to input (backpropagation).

🕸️ 4. Structure of a Fully Connected Neural Network (FCNN)

📥 Input Layer:

- Takes raw features (e.g., pixel values, numerical features like house size, rooms, etc.)

🧠 Hidden Layers:

- Every neuron is connected to **all** neurons in the previous and next layer.
- Responsible for **feature transformation** and learning complex patterns.

📤 Output Layer:

- Final prediction
- Activation depends on task:
    - **Sigmoid** → Binary Classification
    - **Softmax** → Multi-class Classification
    - **Linear or ReLU** → Regression

⚙️ 5. Activation Functions Summary

Layer Type	Common Activation Functions
Hidden Layer	ReLU, Tanh
Output Layer	Sigmoid (binary), Softmax (multi-class), ReLU/Linear (regression)

- **ReLU** is preferred in hidden layers for efficiency and avoiding vanishing gradient.
- **Softmax** ensures output is a probability distribution.

---

6. Applications of Fully Connected Neural Networks

Domain	Use Case
Computer Vision	Digit classification (e.g., MNIST)
Regression	Predict house prices
Speech Recognition	Convert voice to text
NLP	Sentiment analysis, spam detection
Finance	Credit scoring, fraud detection

---

7. Key Takeaways

- **FCNN**: Each neuron in one layer is connected to all neurons in the next.
- **Forward Propagation**: Calculates outputs and loss.
- **Backward Propagation**: Updates weights using gradients.
- **Loss Functions** and **Activation Functions** are chosen based on the type of problem.
- FCNNs are widely used across domains and serve as the **base architecture** for many advanced networks.

Next Steps

In the upcoming lectures, you will:

- Explore advanced architectures (CNNs, RNNs, etc.)
- Learn about regularization (dropout, L2)
- Study optimization techniques (Adam, RMSProp)
- Work on projects applying these concepts