# Deep Learning Workshop

INFO 1998: Intro to Machine Learning

Cornell Data Science

November 20, 2024

# Overview

1. What is Deep Learning?
2. PyTorch Demo
3. Application: Natural Language Processing
4. Conclusion

# A Brief History Lesson

- 1957: **Perceptron** (a.k.a McCulloch-Pitts neuron) first simulated by Frank Rosenblatt at Cornell 🐻
- 1969: Minksy & Papert show perceptron can't learn XOR function $\rightarrow$ AI Winter 😦❄️
- 1986: Rumelhart, Hinton & Williams work popularizes neural networks with hidden layers
- 2020-present: AI Boom! transformers, LLMs (GPT, Llama3 etc.)

# What is Deep Learning?

- Supervised Learning setting
  - Labelled Data: $\mathcal{D} = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\}$
  - Data $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_d^{(i)}] \in \mathbb{R}^d$
  - $d$ features, i.e. data is $d$-dimensional
  - Output is $y_i$, a scalar
- Multiple "layers" which are connected allow model to learn representation of complex data
  - Linear layers = single linear perceptron
  - *Sound familiar?* 👀
- **hyperparameters**: number of layers, size of layer, non-linearity
- **parameters**: matrices of "weights" that are learned during training (used for transforming output of each consecutive layer)

# Neural Network Visual

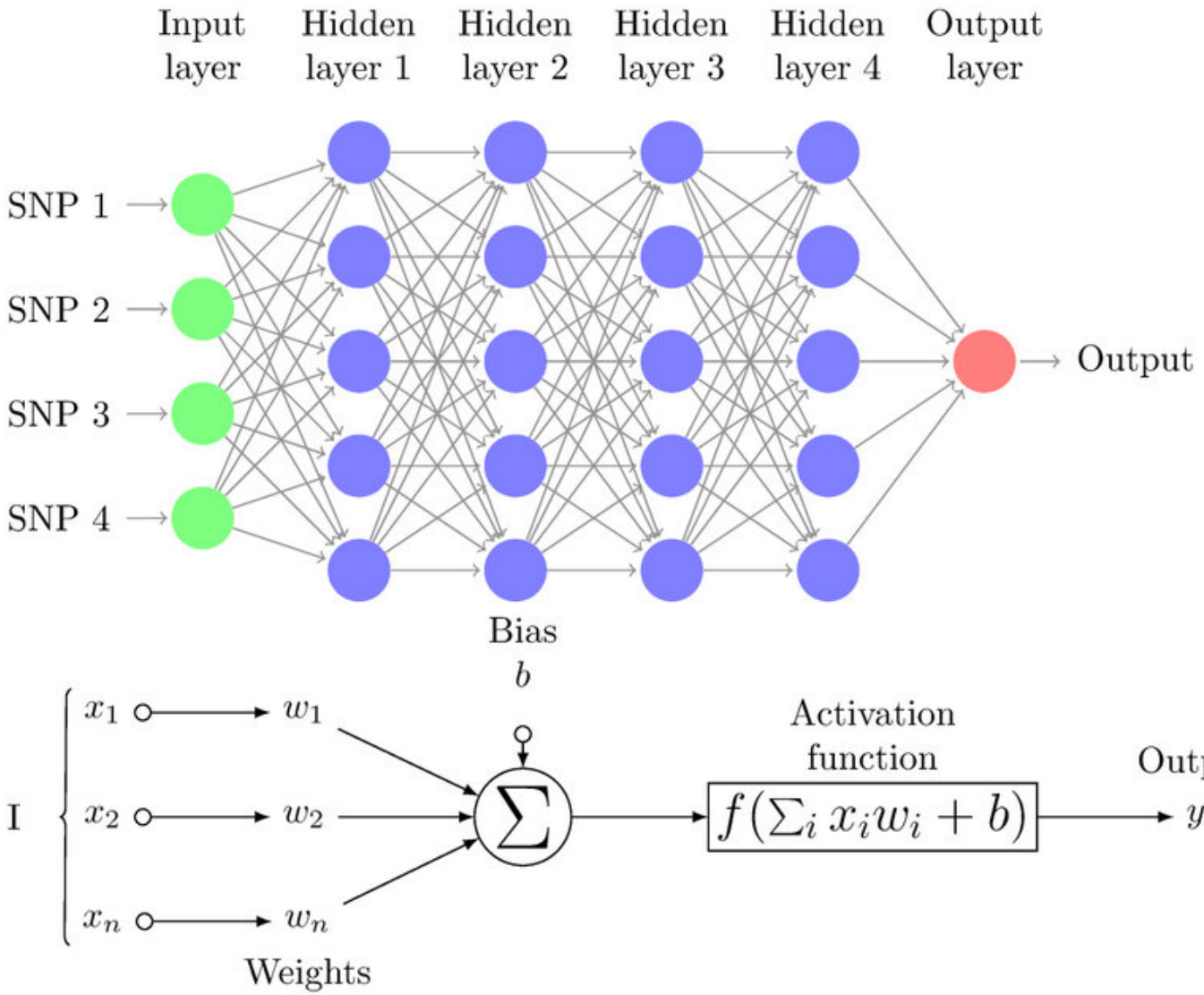


Image Source

# What is Deep Learning? (cont.)

- Interactive Neural Network Demo!
- Neural network with 1 single (infinitely large) hidden layer is a "universal function approximator"
- Need **non-linear** activation function for full expressivity
- Different deep learning model architectures
  - Multi-Layer Perceptron (aka Feed-Forward Neural Networks)
  - Recurrent Neural Networks (RNNs)
  - Long Short Term Memory (LSTMs)
  - Convolutional Neural Networks (CNNs)

# Overview


1. What is Deep Learning?
2. PyTorch Demo
3. Application: Natural Language Processing
4. Conclusion


# Training a Neural Network: An Overview

- Similar procedure as training other models
  - Produce outputs for training data using current model parameters
  - Calculate loss (i.e. compare predictions to ground truth)
  - Update model weights based on loss
- **Forward propagation**
- **Loss function calculation** (i.e. cross-entropy, squared-error loss)
- **Backward propagation** (gradient descent)

# MNIST Dataset Demo

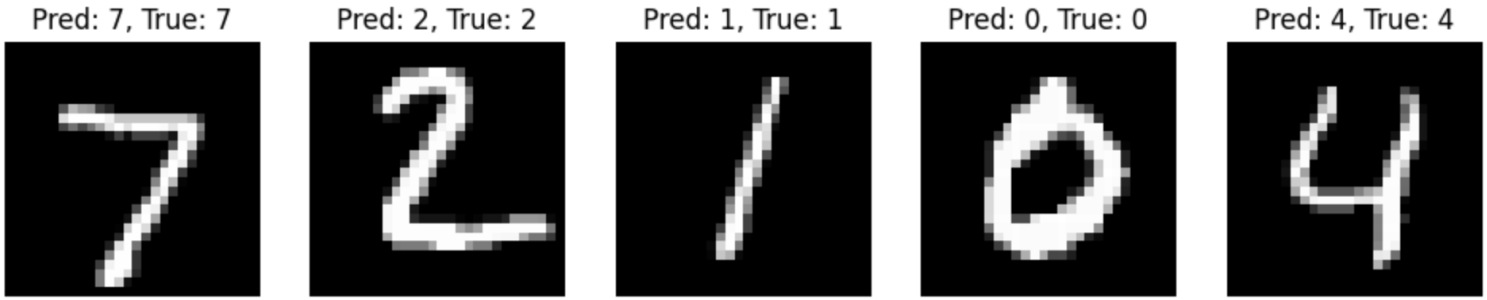


Figure: Classifying Images of Handwritten Numerical Digits

- Google Colab Link!

# Attendance!

Fill this out for extra credit on your attendance score!

# Overview

1. What is Deep Learning?
2. PyTorch Demo
3. Application: Natural Language Processing
4. Conclusion

# Large Language Models (LLMs)

- What all the current hype is about ✨
- LLMs are *neural networks* trained on massive datasets of text to perform natural language processing (NLP) tasks.
- Applications of LLMs
  - Text generation (eg. chatbots)
  - Machine translation
  - Sentiment analysis
  - Summarization
  - Question Answering
- Compute-intensive: High training and inference costs
- Hallucinations: potential for generating inaccurate information

# Overview

1. What is Deep Learning?
2. PyTorch Demo
3. Application: Natural Language Processing
4. Conclusion

# Recommended Resources

- MIT Intro to Deep Learning Lecture Videos
- 3Blue1Brown Deep Learning (mathematical intuition)
- CS 3780: Intro to Machine Learning
- ORIE 3120: Practical Tools for Operations Research, Machine Learning and Data Science
- CS 4782: Intro to Deep Learning *(new Cornell course!)*

# INFO 1998 Final Announcements

- Final Project due **November 26th, 2024** 📝
- Please fill out our End Of Semester Feedback Form
- Thank you to everyone for an awesome semester! 😄
- Keep in touch:
  - Instagram: @cornelldatascience
  - CDS Website
- Q&A Time!