1 Problem 1: Description of the Problem, Subscribe a ML/NN Based Solution

Process:

Definition:

Example:

2 Problem 2: Prescribe a strategy to optimize the NN.

Process:

Definition:

Example:

3 Problem 3: Explain step by step inference for basic algorithms (MLP, CNN, GNN, attention mechanism) in terms of numpy or basic tensor operations.

Process:

Definition:

Example:

4 Problem 4: Be able to explain why such a solution might work or fail.

Process:

Definition:

Example:

5 L2 ML

Summary:

- What does it mean when a computer program learns?
- What is a neural network?
- What is something that a AI model CANNOT do right now?

6 L3 Neural Networks

Summary:

- What is a neural network (NN)?
- How is a GLM and a neural network related?
- What is a representation? Other names for this?
- How do NN make predictions?
- How do NN learn?

7 L4 Neural Network Engineering

Summary:

- What is a MLP?
- What is an inductive bias and why might they be useful?
- What is the difference between hyperparameters and parameters?
- How do we optimize all parameters in a model?

8 L5 Optimizing Hyperparameters

Summary:

- What stategies can help a NN converge when traiing?
- What hyperparameters does a NN architecture have?
- How can we optimize parameters without gradients?
- DL requires a lot of data, what can we do wehn data is scarce?

8.1 Learning representations of data

8.1.1 AI/MI/ML/DL

Definition:

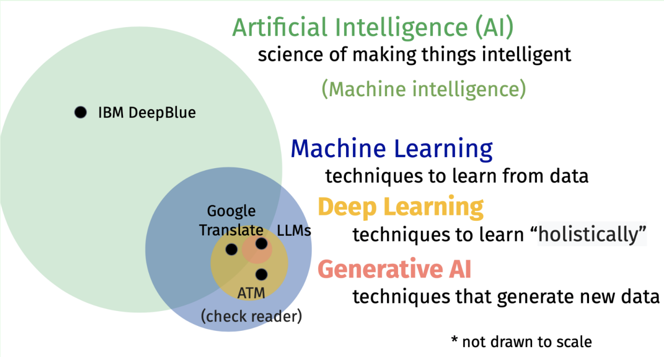

Artificial Intelligence (AI)
science of making things intelligent
(Machine intelligence)
IBM DeepBlue
Machine Learning
techniques to learn from data
Deep Learning
techniques to learn "holistically"
Generative AI
techniques that generate new data
Google Translate
LLMs
ATM (check reader)
* not drawn to scale

Figure 1:

8.1.2 Learning algorithms

Definition: "A **computer program M** is said to **learn** from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at **tasks in T**, as measured by **P**, improves with **experience E**"

- **experience E** $\sim$ Data

- **performance measure P** ~ Loss function, evaluation metric
- **tasks T** ~ "Prediction problem"
- **computer program M** ~ Model
- **learn** ~ Optimize

8.1.3 Linear models

Definition: Linear Regression

$$W \cdot x = y \tag{1}$$

- **E?** (x and y)
- **P?** mean squared error
- **T?** Predict y from x
- **M?** Linear model (W)
- **learn?** Analytical solution or gradient descent

Definition: Generalized Linear models in equations

$$\text{Link}(W \cdot x) = y \tag{2}$$

- x: Input features
- W: Linear transformation
- y: Output / target
- $\text{Link}(x)$: Warping function

Example:

1. If x has dim 50 and W projects to dimension 100, what is the shape of W?
   - W is a 100×50 matrix
2. If W is learnable, how many parameters does W have?
   - $100 \times 50 = 5000$ parameters

Notes: How does a generalized linear model make a prediction? By either mapping to a line or separating data by a line (hyperplane)

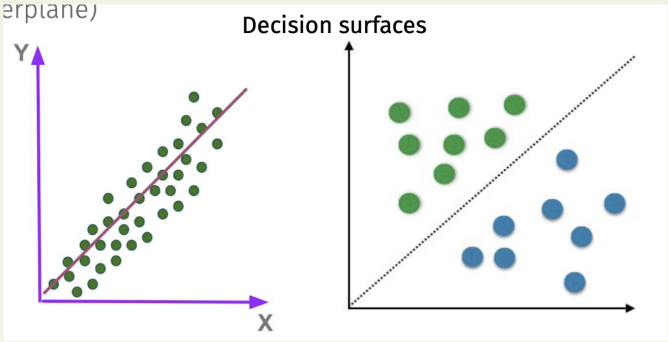


Figure 2:

Notes: What can we do when the data cannot be separated by a line? Resort to different decision surfaces.

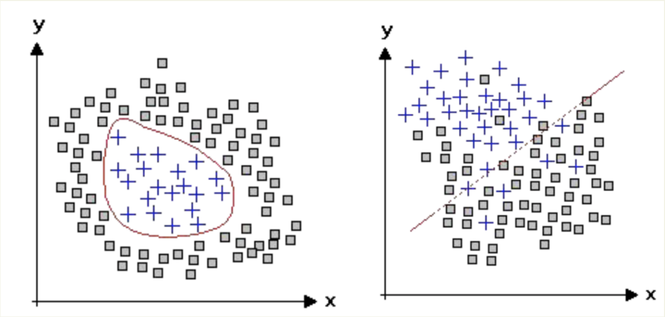


Figure 3:

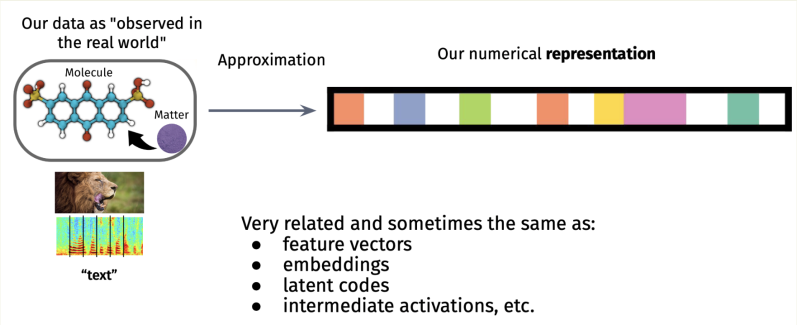
8.1.4 Representations

Definition: **Representation** is a way of encoding data.

$$x \overset{\text{Representation}}{\mapsto} z \tag{3}$$

- z: Feature vectors, embeddings, latent codes, intermediate activations, etc.

Notes:


Our data as "observed in the real world"
Molecule
Matter
"text"
Approximation
Our numerical **representation**

Very related and sometimes the same as:

- feature vectors
- embeddings
- latent codes
- intermediate activations, etc.

Figure 4:

8.2 Neural networks

Definition: Learnable (optimizable) transformations of data.

$$x \overset{\text{Model}}{\mapsto} y \tag{4}$$

8.2.1 2-layer MLP

Definition: By stacking linear transforms with activation functions.

$$\text{Link}(W_2 \cdot \text{relu}(W_1 \cdot x)) = y \tag{5}$$

- x: Input features.
- W_1, W_2: Linear transformations or Weight Matrices.
- $\text{relu}(x) = \max(0, x)$: Non-linear activation function, s.t. $f'(x) = 1$ if $x > 0$ and 0 otherwise.
- y: Output / target.

- **E?** (x and y)
- **P?** mean squared error

- T? Predict y from x
- M? Neural net (W1, W2)
- **learn?** gradient descent

Example:

1. What purpose does relu serve?
   - Introduces non-linearity into the model, allowing it to learn more complex functions.
2. If x has dim 50 and y dim 10, we have layer size of 50, how many parameters do we have?
   - W_1 is a 50×50 matrix, so it has $50 \times 50 = 2500$ parameters.
   - W_2 is a 10×50 matrix, so it has $10 \times 50 = 500$ parameters.
   - Total parameters: $2500 + 500 = 3000$ parameters. IS THIS CORRECT?

8.2.2 Geometric intuition

Notes: **Decision surfaces** Different ways of cutting up space to make predictions.

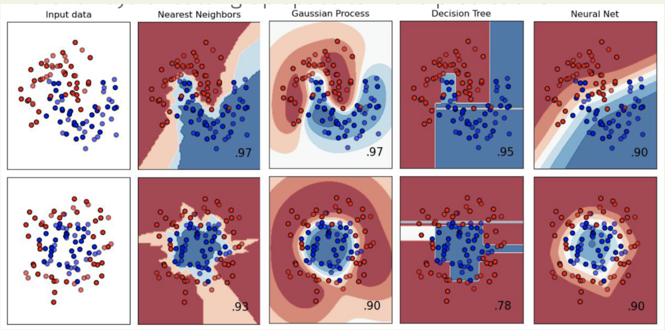


Figure 5:

Notes: **Linear Transformation** Transform data from one vector space to another

$$W \cdot x \tag{6}$$

Notes: **SVD of Linear Transformation** Factorizing matrices into geometrical transformations.

$$W = U\Sigma V^T \tag{7}$$

- U, V: Rotation
- Σ: Scaling

Notes: **Affine Transformation**

$$W \cdot x + b \tag{8}$$

- b: Bias vector

- Translate (b)
- Rotate (W-SVD)
- Reflect (W-SVD)
- Scale (W-SVD)
- Project up or down (dimensionality of $W\mathbf{x}$)

Notes: **ReLU** Rectified linear unit, which has a geometric effect of "gating", some info passes, some doesn't.

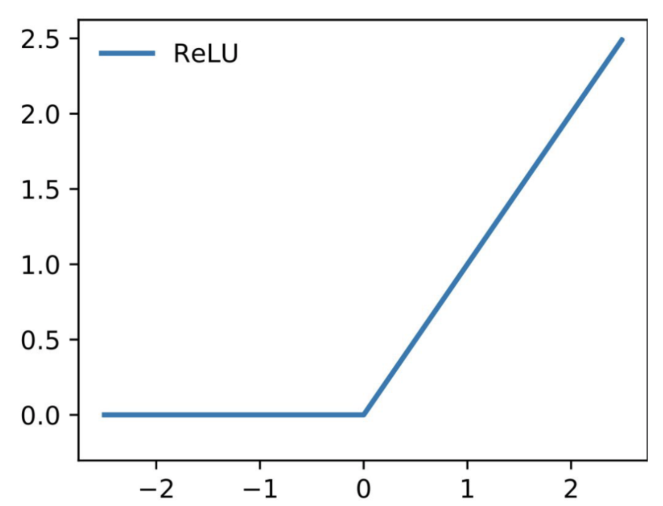


Figure 6:

Notes: **Neural nets** Learn to warp space to make better predictions.

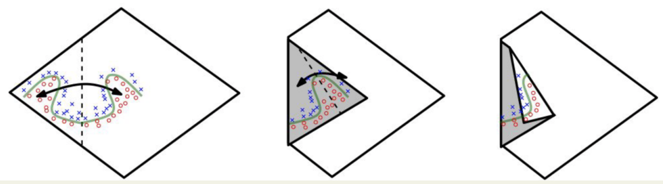


Figure 7:

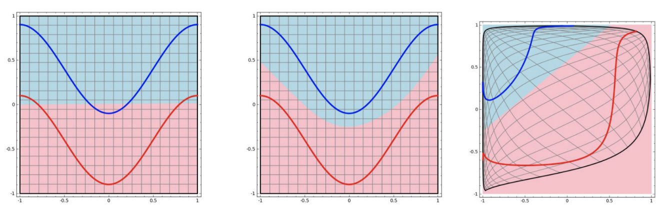


Figure 8:

8.2.3 Encoder-Decoder view

Definition:

$$x \overset{\text{Encoder}}{\mapsto} z \overset{\text{Decoder}}{\mapsto} y \tag{9}$$

- z: Embeddings, latent vectors, learned representations.

Example: **Supervised Learning**

$$x \overset{\text{Model}}{\mapsto} y \tag{10}$$

- $\text{Model}(x) = \text{Decoder}\big(\text{Encoder}(x)\big)$
- $\text{Decoder}(z) = \text{Pred}(z)$

Example: PCA

$$x \overset{\text{Encoder}}{\mapsto} z \overset{\text{Decoder}}{\mapsto} y \tag{11}$$

- Encoder$(x) = W \cdot x$
- Decoder$(z) = W^{-1} \cdot z$

- **E?** x
- **P?** Reconstruction loss
- **T?** Reduce dimension
- **M?** W
- **learn?** Eigendecompositions

Example: Neural Networks

$$x \overset{\text{Encoder}}{\mapsto} z \overset{\text{Decoder}}{\mapsto} y \tag{12}$$

- Encoder(x) = Neural Network
- Decoder(z) = Neural Network

8.2.4 Typical ML Pipeline

Notes:

- Setup data (x, y)
- Define a model: $y = f(x, \theta)$
- Training algorithm to find θ
- Evaluate the model.

8.3 Recap

Summary:

- What is a MLP? Vector-in vector-out optimizable, learnable transformation of data.
- What is an inductive bias and why might they be useful? Set of assumptions that the learner puts on a model for a task, makes an algorithm learn one pattern over another.
  - Let certain patterns be learnable (restricting the hypothesis space).
  - Last layer in the GLM: Restricting output values to 0 and 1.
- What is the difference between hyperparameters and parameters? Hyperparameters are set before training (usually discrete), parameters are learned during training (continuous to learn).
- How do we optimize all parameters in a model? Back propagation.

8.4 Optimizing Hyperparameters

Summary:

9 Representation Learning and Variational Autoencoders

Summary:

- How do we optimize without gradients?
  - Heuristics such as grid search, random search, Bayesian optimization, grad student.
- How does Bayesian Optimization work? (in a nut shell)
  - Model the complex objective function with a surrogate model (GP), and optimize the surrogate model by using the max and min of the acquistion to find the next point to evaluate on the objective function and get the uncertainty of the surrogate model.
- How do we learn on data without labels?
  - Unsupervised learning, self-supervised learning, semi-supervised learning, transfer learning, multi-task learning, meta-learning.

9.1 Principal Component Analysis

Definition: PCA uses linear transformations for dimension reduction.

9.1.1 PCA Limitations

Definition:

1. **Capacity for representation:** PCA finds linear correlations and cannot capture non-guassian structure.
2. **Interpretability:** Other tensor decomposition like non-negative factorization are more interpretable.

10 Autoencoders (AE)

10.1 Non-Linear Dimensionality Reduction

Definition: The encoder maps input to lower dimension and decoder reconstruct the input.

10.2 Loss Function of an AE

Definition: Autoencoders are trained to minimize reconstruction error.

Notes:

- If our data is binary, what loss function should you use? Binary cross entropy.

10.3 Different Types of AEs

10.3.1 Sparse Autoencoders (SAEs): Feature Selection

Definition: SAE encourage latent representations with only few active neurons.

$$\Omega(\theta) = \lambda \sum_i |\theta_i| \tag{13}$$

Notes:

- What regularization technique encourages sparsity? L1 regularization.

10.3.2 Denoising Autoencoders (DAEs): Robust Features

Definition: DAEs enhance robustness by training on noisy inputs

10.4 Anomaly Detection with AEs: Detecting Outliers

Definition: AEs can detect anomalies by analyizing high reconstruction errors.

$$A(\mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{14}$$

$$A(\mathbf{x}) > \text{Threshold} \tag{15}$$

10.5 Latent Space of AEs

Definition: The latent space of an autoencoder may not be continuous

11 Variational Autoencoders (VAEs): A Probabilistic Approach

Summary:

- Work on any data.

Definition: VAEs learn smooth latent spaces by modelling probability distributions.

11.1 VAE's Encoder and Decoder

Definition: The encoder maps input to a distribution, the decoder samples from it.

$$f_{\text{enc},\phi}(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x})\mathbf{I}) \tag{16}$$

Notes:

- $\sigma^2(\mathbf{x})\mathbf{I}$: Diagonal covariance matrix, so it is independent.

11.1.1 Reparameterization Trick

Definition: Reparameterization makes sampling differentiable for VAE training.

11.2 VAE's Loss Function: Reconstruction and KL Divergence

Definition: Balances reconstruction and latent space regularization.

$$L_{\text{VAE}} = L_{\text{AE}}(x, \hat{x}) + \text{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})) \tag{17}$$

11.3 VAE are generative models

Definition: VAE's latent space is continuous and allows sampling new data

11.4 Assignment 1:

Definition: Geometry of high dimensional space.

Notes:

- What is the dimensionality of Fashion-MNIST? Higher bounded is 784 because it is a 28x28 image, but the lower bound is ill-definied (many answers)
- If it was a colour image, then it would be $3 \times 28 \times 28 = 2352$.

Testng