Natural Language Processing

Neural Networks II: How It Learns

Ramaseshan Ramachandran

OBJECTIVE FUNCTION

- Provides information related to how well the model is in sync with the original data or gold standard
- Refers to a function used for optimization in a machine learning model, whether through minimization or maximization.
- In the context of machine learning, it usually refers to minimizing errors
- In reinforcement learning, it could refer to maximizing the reward

LOSS FUNCTION I

Consider the prediction of y, given x

$$\hat{y} = w^T x + w_0 \tag{1}$$

where $\hat{y}$ is the predicted value
w_0 is the bias
x is the input vector w is the weight vector
If $y$is the target (0 or 1) and $\hat{y}$ is the predicted probability of y being 1 (a value between 0 and 1), then the loss function is:

LOSS FUNCTION II

$$L(y, \hat{y}) = -[y \log(\hat{y}) + (1-y) \log(1-\hat{y})] \tag{2}$$

Minimize **cross-entropy loss**, which measures the difference between predicted and true class distributions. When L is negligible, we confidently predict the correct class x. $L \rightarrow 0$

- The model's prediction is very close to perfect.
- If the true sentiment is positive ($y = 1$), and L is close to zero, then is very close to 1.
- If the true sentiment is negative ($y = 0$), and L is close to zero, then is very close to 0.

L and Confidence

- A low L value also indicates high confidence in its prediction.
- A high L value indicates its low confidence in predicting the label

COST FUNCTION

Lets assume we have vectors for training. In sentiment analysis, these are vectors from word embedding methods, representing sentiment words. We could also use one-hot vectors.

$$X = [x_1, x_2, x_3, \ldots, x_N] \tag{3}$$

Combining the model parameters $w_0, w_1, w_2, w_3, \ldots, w_n$ with the loss function, we get a **Cost function**, averaged over all the input training samples

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y_i}) \tag{4}$$

- The loss is a function of prediction and target values of a single training example
- The cost is a function of model parameters and bias - average of the loss over all training samples

COST FUNCTION I

Let's assume we have a set of training vectors, denoted as:

$$X = [x_1, x_2, x_3, ..., x_N] \tag{5}$$

- where eachx_i represents a data point. In sentiment analysis, these vectors could be obtained from word embedding methods (e.g., Word2Vec, GloVe) to represent sentiment words, capturing semantic relationships. Alternatively, one-hot vectors could be used, although they are less informative.
- The Cost Function is a function of the model parameters, denoted as θ, and is obtained by averaging the loss function over all training samples. It provides an overall measure of how well the model is performing on the entire training set.

COST FUNCTION II

A common form of the cost function, using the squared error loss, is:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y}_i) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{6}$$

- where N is the total number of training samples.
- y_i is the true sentiment label for the i^{th} sample.

  $\hat{y}_i$ is the predicted sentiment label for the i^{th} sample.
- The choice of loss function depends on the specific nature of the problem and the desired behavior of the model.
- Additionally, the cost function may include **regularization terms** to prevent over fitting and encourage simpler models.

GRADIENT DESCENT

- GD iteratively used to adjust the weights and as a result to minimize the cost function
- Initialize the weights to random values
- Iteratively adjust the weights in the direction of the steepest descent or in the direction that most decreases the cost function. To update the weights in the steepest descent, a learning parameter η is used

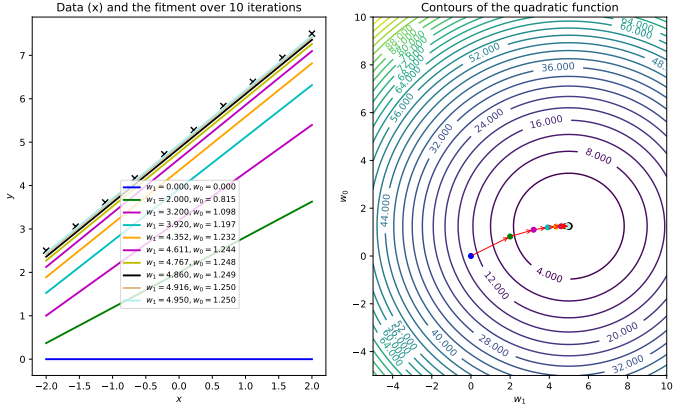
$$w_j \leftarrow w_j - \eta \frac{\partial J(\theta)}{\partial w_j} \tag{7}$$

$$= w_j - \eta \sum_{i=1}^{N} x_j^{(i)} (\hat{y}^{(i)} - y^{(i)}) \tag{8}$$

where η is the learning rate, typically between 0.001 and 0.01

GD: ADVANTAGES AND LIMITATIONS

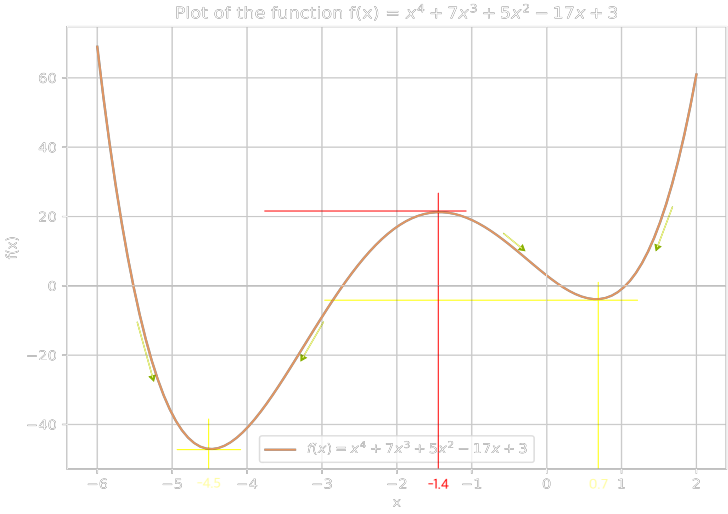
- Iterative
- Computationally efficient
- Generic and could be used to solve even non-linear equations
- Suitable for large models
- It works!
- It is very slow when it reaches close to the local minima



CONTINUOUS OPTIMIZATION[**Deisenroth_Faisal_Ong_2020**]

- Numerical methods to train machine learning models
- To find a good set of parameters
- The objective function or probabilistic model determines what counts as a good set of parameters
- Objective functions are assumed to be differentiable
- In most cases, the objective functions are minimized - the best value is the optimum value/minimum value
- Finding the point or points (ideally deepest point) that have zero gradient
- Move in the direction opposite to the gradient (which points uphill) to reach the minimum value

EXAMPLE OF A FUNCTION



STEP SIZE

- Choosing a step size is important
- If the step size is too small, gradient descent will be slow
- If the step size is large, gradient descent can oscillate, overshoot, fail to converge, or even diverge
- Adaptive step size (changing it every iteration) - based on the local properties of the function
  - When the function value increases after the gradient descent, reduce the step size
  - If the function value decreases, you may increase the step size

SOLVING A SYSTEM OF LINEAR EQUATIONS

- Consider the equation $Ax = b$
- Find x by minimizing the squared error:

$$\|Ax - b\|^2 = (Ax - b)^T (Ax - b)$$

The gradient with respect to x is: $\nabla f(x) = 2A^T(Ax - b)$

- The rate of convergence depends on the condition number $\kappa = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$
- Lower κ leads to faster convergence and fewer iterations required to reach a desired value
- Higher κ (poor conditioning) means slower convergence and more iterations

INTRODUCTION TO LOGISTIC REGRESSION

- **Logistic regression** is a classification algorithm used to predict the probability of a binary outcome.
- We want to model the conditional probability $p(y = 1|x)$ as a function of x for the binary output variable y and making $p(.)$ be a linear function of x won't work
- Modelling $\log p(x)$ as linear is unbounded in one direction; modelling $\log \frac{p}{1-p}$ works. This quantity is known as the ***logit*** (log-odds)
- The prediction is based on the logistic function (sigmoid):

$$P(y = 1|x) = \frac{1}{1 + e^{-w^T x}}$$

  where x is the input vector, w is the parameter vector.
- Logistic regression outputs a probability and classifies based on a threshold (usually 0.5).

SEQUENTIAL NATURE OF DATA

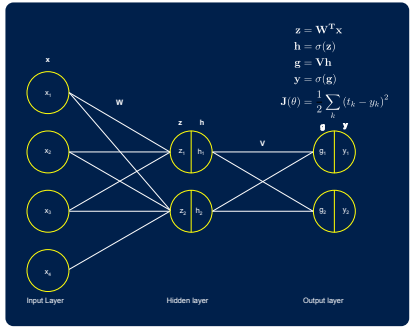
- Speech
- Documents
- Videos
- Weather forecast
- Financial - Stock market

PROPERTIES OF ANN

- Massively parallel distributed structure
- Ability to learn
- Ability to learn from training samples
- Ability to find latent patterns in the data
- Generalize and associate data

BACKPROPAGATION MODEL

The goal of backpropagation is to change the weights so that the *estimated target* $\approx$ target, thereby minimizing the error for each neuron and the network as a whole.

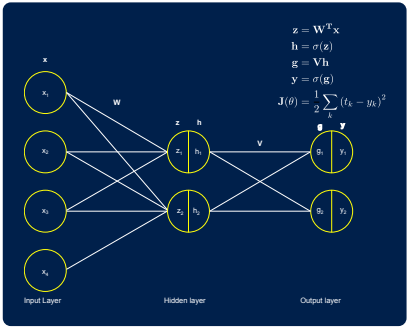


The goal is to minimize

$$J(\theta) = 1/2\Big((t_1 - y_1)^2 + (t_2 - y_2)^2\Big) \tag{9}$$

* We want to the adjust weights coming in and going out of hidden layer so that $t - y$ is minimized
* $\Delta W \propto -\dfrac{\partial J(\theta)}{\partial W}$

BACK PROPAGATION MODEL - FORWARD PASS



$$z_1 = w_1^T x + b_1 \tag{10}$$

$$z_2 = w_2^T x + b_2 \tag{11}$$

$$z_1 = x_1 w_{11} + x_2 w_{21} + \cdots + b_1 \tag{12}$$

$$z_2 = x_1 w_{12} + x_2 w_{22} + \cdots + b_2 \tag{13}$$

$$h_1 = \sigma(z_1) = \frac{1}{1+e^{-z_1}} \tag{14}$$

$$h_2 = \sigma(z_2) = \frac{1}{1+e^{-z_2}} \tag{15}$$

$$g_1 = h_1 v_{11} + h_2 v_{21} \tag{16}$$

$$g_2 = h_1 v_{12} + h_2 v_{22} \tag{17}$$

$$y_1 = \sigma(g_1) = \frac{1}{1+e^{-g_1}} \tag{18}$$

$$y_2 = \sigma(g_2) = \frac{1}{1+e^{-g_2}} \tag{19}$$

GENERALIZATION

Most living beings have the ability to generalize or transfer information from previous experience so that they can behave appropriately in novel situation

- Global shape
- Local features
- Physical characteristics

  shape, Size, doors, windows, wheels, etc.
- Recognition from Different Angles
- Prior Knowledge and Experience
  - Pattern completion or fill-in capabilities

Combination of cues, both visual and contextual, allows humans to identify a bus from all directions and with either partial or full views in the 3-D space or in 2-D drawings

ADVANCED PERSPECTIVES ON GENERALIZATION

- Generalization is crucial in both machine learning and scientific research
- It enables computed models or theories to apply knowledge to unseen contexts
- This is vital for predictive accuracy beyond training data.

A BROADER VIEW

- Involves formulating universal principles based on specific observations
- Aims to predict phenomena in new scenarios without needing additional evidence for every case
- Predict real-world complexities, ensuring predictions are applicable/meaningful across a wide array of situations

GENERALIZATION IN MACHINE LEARNING

- Refers to a models ability to predict outcomes on unseen data - not just from the training set
  - Captures the underlying patterns of the data rather than just memorizing the training examples
- Refers to true effectiveness to predict the outcome beyond training performance

TRAINING VS TESTING

- Models are trained on labeled data to find patterns
- Weights and biases model the relationship between inputs and outputs
- The linear combination of features followed by a non-linear activation function creates a generalized model

$$h_i = \sigma\left(\sum_{j=1}^{n} w_{ij} x_j + b_i\right)$$

- Non-linearity enables the model to learn complex patterns and relationships that cannot be captured by just linear combinations
- This non-linearity is crucial for generalization, as it allows the model to fit a wider range of data distributions

FEATURE ABSTRACTION

- Each neuron learns to recognize specific patterns or combinations of features in the data. The neurons in the hidden layer learn and abstract complex features from textual data
- As the number of hidden layers increases, the intuition is that they capture more nuances of the text, including grammar, meaning, context, and intricate patterns.
  - A pattern such as spill the beans is associated with its idiomatic meaning rather than the straightforward literal one
  - The sarcasm in this sentence Wow, they improved by a whopping 4 runs this time from their lowest! cannot be captured using a single layer of hidden units
  - The word **wow** is usually present in a positive context, but here it does not refer to the positive sentiment
  - A model that relies only on linear operations wouldn't capture these kinds of nuances, as it would attempt to separate words and meanings using linear decision boundaries

TRANSFORMATION OF INPUT DATA

- Hidden layers serve as transformation layers that convert input textual data into abstract representations
  - GloVe and Word2Vec captures semantic similarities and context-aware representations
  - Hidden layers facilitate hierarchical learning where each layer captures increasingly abstract features
  - Each hidden layer builds upon the previous layers learned features - brings in the deep leaning capabilities

REPRESENTATION LEARNING I

- The concept of representation learning highlights that hidden layers learn to represent input data through transformations that reflect its underlying structure.
- **Contextual Understanding** - Modern NLP models like Transformers learn representations that incorporate context, allowing a word to have different embeddings based on its surrounding words, improving handling of polysemy and ambiguity.
- **Transfer Learning** - Pre-trained language models (e.g., BERT, GPT) are used as a base, and their learned representations are fine-tuned on specific downstream tasks, improving generalization to new NLP problems with minimal task-specific data.
- **Hierarchical Feature Learning**- Neural networks with multiple layers (especially deep models) build hierarchical representations, where initial layers capture lower-level features (like word patterns), and deeper layers capture more abstract semantics, improving model performance on complex tasks.

REPRESENTATION LEARNING II

- **Unsupervised Learning**: Representation learning often leverages unsupervised or self-supervised techniques (e.g., masked language models), enabling models to learn representations from raw text data without requiring labeled examples.

OVERFITTING AND UNDERFITTING

- **Overfitting** occurs when a model memorizes the training data, including noise and outliers, leading to poor performance on unseen data.
- **Underfitting** happens when a model fails to learn significant patterns from the training data, causing it to perform poorly on both training and test sets.

GENERALIZATION TECHNIQUES

- **Regularization**: Methods like L1 or L2 regularization introduce penalties for model complexity, reducing overfitting.
- **Cross-Validation**: Splits the dataset into subsets to ensure consistent performance across different samples.
- **Data Augmentation**: Expands the dataset through transformations, helping models learn more robust features by introducing variation in the data.

REFERENCES I