CS 419M Introduction to Machine Learning **Spring 2021-22**

Lecture 13: Practical Machine Learning and Non-linearities

Lecturer: Abir De *Scribe: Group 1*

13.1 Practical Machine Learning

13.1.1 Reason for the sudden burst in nonlinear models

Machine Learning has its foundations in **Statistics** and Statistics mainly dealt with linear models because of their simplicity and theoretical guarantees that are associated with them. However, it is a well-known fact that most data is nonlinear. A paper by George Cybenko in 1989 demonstrated that a neural network with arbitrary width could approximate any continuous function with sigmoid activations.

In the past few years, Neural Networks have become the go-to models for any ML task. The reason for this can be attributed to the development of software libraries like **Tensorflow** and **PyTorch** that made gradient computations convenient. This allowed researchers and practitioners to focus on model development.

13.1.2 Nonlinearities in Regression

Consider $y = f(x)$ where $x \in \mathbb{R}$ is a data instance defined by a set of features. f maps x to the target, y. f is not known to us and we must infer f from the training set, S. A concrete example is $y = \log(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Polynomials are the simplest nonlinear models to work with and we can define,

$$y = f_\theta(x) = \sum_{i=1}^{N} \theta_i x^i \tag{13.1}$$

However, $\lim_{x\to 0} \log(x) = -\infty$ while $\lim_{x\to 0} f(x) = 0$. This can be resolved by including negative powers of x,

$$y = f_\theta(x) = \sum_{i=-N}^{N} \theta_i x^i \tag{13.2}$$

This model suffers from the problem that as i increases, θ_i also increases.

13.1.3 Universal Approximators

Although polynomials are simple models to work with, they are **not** universal approximators.On the other hand, the Linear-ReLU-Linear (**LRL**) is a universal approximator

Put more formally, *given any function f, $\forall \epsilon > 0$, $\exists$ an LRL layer parametrised by θ such that $\|f - LRL_\theta\| < \epsilon$ where $\|\cdot\|$ is a valid norm.*

The LRL layer can be used to predict the target $\mathbf{y}$ using $\mathbf{X} \in \mathbb{R}^{|X|}$ as follows:

$Z = WX + b$, $W \in \mathbb{R}^{|Z| \times |X|} \longrightarrow Z' = ReLU(Z) \longrightarrow y = W_1 Z' + b'$, $W_1 \in \mathbb{R}^{1 \times |Z|}$

Model Complexity can be increased by:

1. Increasing the dimension of Z (*dim(Z)*)
2. Cascading a large number of LRL blocks in series

$$\mathbf{X} \longrightarrow \boxed{\text{LRL}} \longrightarrow \boxed{\text{LRL}} \longrightarrow \boxed{\text{LRL}} \longrightarrow \boxed{\text{LRL}} \longrightarrow \cdots \longrightarrow \boxed{\text{LRL}} \longrightarrow \mathbf{y}$$

Although promising, in going from linear to nonlinear models, we give up on the ability to obtain closed form solutions, which are easy to visualize and compute. The absence of closed-form solutions leads us to devise algorithms, one of which is Gradient Descent.

13.2 Gradient Descent

Since nonlinear models do not permit closed-form solutions, we adopt an **iterative approach** to train our ML models. **Gradient Descent** is an iterative first-order optimisation algorithm used to find a local minimum/maximum of a given function.

13.2.1 Setup

Our proposed model is, $y \approx f_\theta(x)$ where f is parametrised by $\theta \in \mathbb{R}^n$.
For general function f we have to minimize,

$$\min_{\mathbf{f}} \sum_{i \in \mathcal{D}} (y - f(x_i))^2$$

Assuming our algorithm for the model is working we say,$f \to f_\theta$. Define a loss function, L_θ which is to be minimised. For example, recall the regression loss function,

$$L_\theta = \min_\theta \sum_{i \in \mathcal{D}} (y - f_\theta(x_i))^2$$

The Gradient Descent update rule can be defined as follows,

$$\theta_{t+1} = \theta_t - \gamma \frac{\partial L_\theta}{\partial \theta} \tag{13.3}$$

for a suitable learning rate, $\gamma \in \mathbb{R}$.
What does suitable learning rate mean?Since the gradient gives the *direction of steepest ascent*, Gradient Descent updates the parameters in the direction of steepest descent. In practice, γ must be chosen carefully as too large a learning rate would result in the loss function increasing, rather than decreasing. A theoretical bound is given for the value of γ, assuming a convex loss function.

13.2.2 Proof of Convergence for Convex Losses

Theorem 13.1. *Consider the optimisation problem,* $\min_\theta L(\theta)$ *where* L *is given to be convex with respect to* θ. *If* $\gamma < \frac{2}{\lambda}$ *where* $\lambda = \max eigs(\mathbb{H}(\theta))$, *then Gradient Descent converges to the global optima of* L.

Proof. Using the *Taylor Series* expansion of L in a neighborhood of θ_t,

$$L(\theta) = L(\theta_t) + \left(\frac{\partial L}{\partial \theta}\right)^T\Bigg|_{\theta_t} (\theta - \theta_t) + O(\theta^2)$$

For convex losses,

$$L(\theta) - L(\theta_t) \geq \left(\frac{\partial L}{\partial \theta}\right)^T\Bigg|_{\theta_t} (\theta - \theta_t)$$

For $\theta = \theta_{t+1}$:

$$L(\theta_{t+1}) - L(\theta_t) \geq \left(\frac{\partial L}{\partial + \theta}\right)^T\Bigg|_{\theta_t} (\theta_{t+1} - \theta_t)$$

But,

$$L(\theta_{t+1}) - L(\theta_t) \leq \left(\frac{\partial L}{\partial \theta}\right)^T\Bigg|_{\theta_t} (\theta_{t+1} - \theta_t) + \frac{\lambda_{max}}{2}||\theta_{t+1} - \theta_t||^2$$

where λ_{max} is the maximum eigenvalue of the Hessian. From gradient descent, we have:

$$\theta_{t+1} - \theta_t = -\gamma\left(\frac{\partial L}{\partial \theta}\right)\Bigg|_{\theta_t}$$

Substituting the value of $\theta_{t+1} - \theta_t$:

$$L(\theta_{t+1}) - L(\theta_t) \leq \left(\frac{\partial L}{\partial \theta}\right)^T\Bigg|_{\theta_t} \left(-\gamma\left(\frac{\partial L}{\partial \theta}\right)\Bigg|_{\theta_t}\right) + \frac{\lambda_{max}}{2}\gamma^2\left|\left|\frac{\partial L}{\partial \theta}\right|\right|^2$$

$$L(\theta_{t+1}) - L(\theta_t) \leq \left(-\gamma + \frac{\lambda_{max}}{2}\gamma^2\right)\left|\left|\frac{\partial L}{\partial \theta}\right|\right|^2$$

For $\gamma \leq \frac{2}{\lambda_{max}}$, we have:

$$L(\theta_{t+1}) - L(\theta_t) \leq 0$$

□

Note that we don't want to use a large value of λ (even it is possible) because a large value of λ may increase the loss instead of decreasing it.

NOTE: *Although we proved the convergence of Gradient Descent for convex losses and found an upper bound on the learning rate, γ, one can also find a similar bound for non-convex losses. Given λ such that $\mathbb{H}(\theta) \leq \lambda \mathbb{I}$ and if $\gamma < \frac{2}{\lambda}$, then Gradient Descent will converge to a local optima.*

However, the inequality $\theta_{t+1} \geq \theta_t - \gamma \|\frac{\partial L_\theta}{\partial \theta}\|^2$ requires convexity.

Even though the same result holds for non-convex losses, γ may be low due to which the parameters are update slowly. Adding an L^2 regulariser, $\frac{a}{2\gamma}\|\theta\|^2$ changes the update rule to,

$$\theta_{t+1} = (1-a)\theta_t - \gamma \frac{\partial L_\theta}{\partial \theta}$$

This ensures that $\theta_{t+1} \neq \theta_t$, preventing the learning algorithm from getting trapped in suboptimal local minima and allows it to explore the optimisation landscape more fully.

Q. ***In practice, should the loss function be divided by the batch size?***
Ans. ***No****. Dividing the loss function by batch size reduces the effective learning rate which slows down model training. This can be avoided by scaling γ appropriately but this is avoided in practice since typical learning rates are supposed to be small.*

13.2.3 Formulation of the Update Rule as an Optimisation Problem

Proposition 13.2. *The Gradient Descent Update rule is the solution to the optimisation problem,*

$$\theta_{t+1} = \arg\min_{\theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \lambda \left(L_\theta + (\theta - \theta_t)^T \frac{\partial L_\theta}{\partial \theta} \right)$$

Proof. The proof follows from differentiating the expression and setting it to zero.

From the definition of the optimisation problem, it is clear that λ acts like a weighting parameter deciding the importance of remaining at θ_t or updating it using a linear approximation of L_θ in the neighborhood of θ_t.

□

13.3 Group Details and Individual Contribution

Section	*Contributor*
13.1.1, 13.1.2	*Krishnasya Arush Tadikonda (190100066)*
13.1.3	*Ankith R (200070006)*
13.2.1	*Bodke Pavan Vijay (200070014)*
13.2.2	*Shashwat Gupta (200070075)*
Notes, 13.2.3	*Aditya Sriram (200070004)*