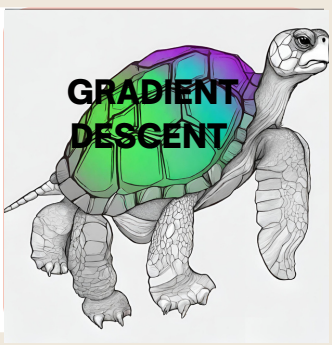
ML OPTIMIZERS WHEN TO USE?

1

GRADIENT DESCENT (GD) IS A BASIC OPTIMIZATION ALGORITHM THAT UPDATES MODEL PARAMETERS BASED ON THE NEGATIVE GRADIENT OF THE LOSS FUNCTION. IT IS SUITABLE FOR SMALL DATASETS AND SHALLOW NETWORKS, AND IS APPLIED BY UPDATING MODEL WEIGHTS BASED ON THE GRADIENTS OF THE LOSS FUNCTION WITH RESPECT TO THE WEIGHTS.

Explanation: The humble knight, loyal and steady, but a bit slow in the realm of optimization. 🐢

- Visual: Slow and Steady Tortoise 🐢
- A tortoise with a gradient descent shell, showcasing the slow but steady progress.



2

Adam (Adaptive Moment Estimation) combines momentum and RMSProp, making it popular for various problems. It is efficient for training deep neural networks and is a common choice due to its adaptive learning rates. Apply it by combining the benefits of AdaGrad and RMSProp.

- Explanation: The All-Seeing Wizard 🧙
- Visual: An image of a wizard with a crystal ball (Adam optimizer) foreseeing the optimal path.

3

STOCHASTIC GRADIENT DESCENT (SGD) IS A POWERFUL OPTIMIZATION TECHNIQUE THAT USES A RANDOM SUBSET OF DATA (MINI-BATCH) TO UPDATE MODEL PARAMETERS IN THE DIRECTION OF THE NEGATIVE GRADIENT OF THE LOSS FUNCTION. SGD IS PARTICULARLY USEFUL FOR LARGE DATASETS, QUICK TRAINING, AND NON-CONVEX OPTIMIZATION. TO APPLY, PERFORM WEIGHT UPDATES FOR EACH MINI-BATCH TO CONVERGE FASTER THAN TRADITIONAL GRADIENT DESCENT.

- Explanation: Scatterbrained Squirrel 🐿️
- Visual: An image of a squirrel scattering nuts (representing data points) in random directions.

4

ADAGRAD IS AN OPTIMIZATION ALGORITHM THAT ADAPTS THE LEARNING RATE DURING TRAINING, ALLOWING FOR LARGER UPDATES FOR INFREQUENT PARAMETERS AND SMALLER UPDATES FOR FREQUENT PARAMETERS. IT'S SUITABLE FOR SPARSE DATA OR DATA WITH INFREQUENT FEATURES, MAKING IT COMMONLY USED IN NATURAL LANGUAGE PROCESSING TASKS.

- Explanation: Learning Librarian 📚
- Visual: A librarian organizing books (representing weights) based on their frequency of use.

 # MAKE LIFE EASY, APPLY THE OPTIMIZERS ACCORDING TO THE NEED, DEPICTED IN THE EXAMPLES ABOVE.

 # NO FEAR WHEN THESE CAPTIONS ARE HERE

1.GD

CAPTION: "Optimization is all about channeling your inner tortoise - taking it slow and steady, with eyes firmly fixed on the prize! 🐢💨 #OptimizationJourney"

2.SGD

CAPTION: "Get ready to embrace the thrill of the unknown! SGD dances through the jungle of data, just like a squirrel searching for its next snack! 🌰🌲 #MLAdventures"

3. ADAM

Behold Adam, the sorcerer of optimization, gazing into the mystical gradients crystal ball to conjure the ultimate spell! ✨🔮 #MagicalOptimization

4. ADAGRAD

Welcome to Adagrad, the learning rate librarian extraordinaire! Let's tidy up our weighty tomes and create the ultimate library of knowledge! 📚🔍 #SmartLearning

Gradient Descent (GD):
Formula: 🐢 Slow & Steady Wins: $\theta = \theta - \alpha * \nabla J(\theta)$
Explanation: The wise tortoise (🐢) takes it slow and steady, following the gradients (∇) to reduce costs (J).

Stochastic Gradient Descent (SGD):
Formula: 🐿️ Randomize & Chase: $\theta = \theta - \alpha * \nabla J(\theta, x(i))$
Explanation: The scatterbrained squirrel (🐿️) bounces around randomly to reduce costs (J) for each data point (x(i)).

Adam:
Formula: 🧙 Wizardly Magic: $m = \beta_1 * m + (1 - \beta_1) * \nabla J(\theta)$ & $v = \beta_2 * v + (1 - \beta_2) * (\nabla J(\theta))^2$ & $\theta = \theta - \alpha * m / (\sqrt{v} + \varepsilon)$
Explanation: The magical wizard (🧙) Adam combines the wisdom of moments (m & v) to guide the steps magically for optimal results.

Adagrad:
Formula: 📚 Adaptive Learning: $G = G + (\nabla J(\theta))^2$ & $\theta = \theta - (\alpha / \sqrt{G + \varepsilon}) * \nabla J(\theta)$
Explanation: The learning librarian (📚) adjusts the step size based on the accumulation of gradients over time (G).