

NLP – Fine tuning(LoRA) Projesi

Projenin Özeti

- Base Model: Qwen2.5-Coder-1.5B-Instruct (1.5B params)
- Method: LoRA Fine-Tuning
- Datasets: Naholav/CodeGen-Diverse-5K, Naholav/CodeGen-Deep-5K
- Training Field: input , solution (code-only, no reasoning)

Yaklaşım

Her bir dataset için ayrı ayrı modelin eğitilmesi ve bu eğitimlerin sonucunda modelin çıktılarının test edilmesi. Sonrasında eğitim sonuçlarına göre yeniden daha doğru ve modele uygun şekilde parametrelerin güncellenmesi ile modelin daha doğru eğitilmeye çalışılması.

Model her iki dataseti için 2'şer kere eğitilmiştir , ilk eğitimde 2 dataset de aynı parametreler ile eğitilmiştir. Sonrasında ikinci eğitimde ilk eğitiminin sonuçlarına bakılarak bazı parametrelerin güncellemesi ile yeniden iki dataset üzerinden eğitilmiştir.

Kullanılan Parametreler ve İşlevleri;

max_seq_len : Modelin attention matrisinin boyutunu belirler. Yani modelin tek seferde okuyacağı metnin uzunlu denebilir. Bellek penceresini belirler, yani model aynı anda kaç token görebilir ve bağlam kurabilir. Bu parametrenin değeri VRAM ile doğru orantılıdır. Ayrıca verilecek olan verinin uzunluğu ile de doğru orantılıdır.

Lr(learning rate) : Modelin bir adımda ağırlıkları ne kadar değiştireceğidir. Yani lora matrisinin(A ve B) güncellenme hızını belirler. Eğer lr büyüğse model hızlı öğrenir ama karasız olup saçmalayabilir. Eğer lr küçükse model daha iyi genel olarak performans artar ve overfit riski azalır fakat eğitim çok yavaş olabilir. Yani çok büyük olursa model bozulur, çok küçük olursa model öğrenemez.

R(rank) : LoRA nın eklediği matrislerin boyutunu belirler. Doğrudan temsil eder. R ne kadar büyüğse LoRA o kadar fazla bilgiyi encode eder, küçükse de çok sınırlı bir değişiklik yapar. Yani R büyükdükçe ana model daha fazla değişime ve öğrenmeye maruz kalır. Ayrıca R, VRAM ile doğrudan ilişkilidir(bire bir). Aynı şekilde R büyündükçe daha fazla parametre olacağı için eğitim hızı yavaşlar, azalırsa da hız artar.

Alpha(a) : LoRA ile yapılan güncellemelerin modele ne kadarlık bir ölçekte etki edeceğini belirleyen katsayıdır. Yani Lora'nın temel modele uygulayacağı güncellemenin güç seviyesidir. Alpha büyündükçe model daha hızlı öğrenir, overfitting riski artar. Alpha küçüldükçe overfitting riski azalır fakat model domain bilgisini yeterince alamayabilir.

Checkpoints : Eğitim sırasında modelin LoRA ağırlıklarının ve optimizer durumunun kaydedilmiş halidir. Böylece eğitim süresince bir versiyonlama gibi her belli bir adımda çıktısı alınmış olur. Ayrıca bir şekilde gpu kesilirse veya bir aksilik olursa diye yedekli bir şekilde kaydetme de denebilir.

Epoch : Eğitim verilerinin tamamının modelden geçirilme sayısını belirler. Yani mesela 10000 tane veri varsa ve epoch 10 seçilmişse , her epoch da 1000 training step olur. Eğer epoch sayısı fazla olursa model ezberler(overfitting), az olursa öğrenemez. Ayrıca epoch sayısı direkt modelin eğitim süresini de etkiler. Fakat VRAM bu değerden etkilenmez.

Her eğitim için kullanılan parametreler:

1. Eğitim;

Parametreler	Deep - Diverse
MAX_SEQ_LEN	1024
BATCH_SIZE	8
LR	2e-4
NUM_EPOCH	3
WARMUP_RATIO	0.03
LORA_R	16
LORA_ALPHA	32
LORA_DROPOUT	0.05
SAVE_STEPS	200
LOG_STEPS	20
SEED	42
TRAIN_RATIO	0.95

2. Eğitim ;

Parametreler	Deep - Diverse
MAX_SEQ_LEN	1024
BATCH_SIZE	8
GRAD_ACCUM_STEPS	2
LR	1.5e-4
NUM_EPOCH	4
WARMUP_RATIO	0.05

LORA_R	32
LORA_ALPHA	64
LORA_DROPOUT	0.08
SAVE_STEPS	200
LOG_STEPS	20
SEED	42
TRAIN_RATIO	0.95

Yapılan LoRA eğitimlerinin sonucunda loglanan değerlerin grafikleri

DEEP – 1;



DIVERSE –1;



DEEP -2;



DIVERSE -2 ;

