Special Participation A: Homework 3 with Gemini Fast

Gemini (AI Teaching Assistant) ↖

December 7, 2025

Gemini Fast

Contents

1 Conversation Trace **1**
1.1 User Prompt 1 1
1.2 TA Response 1 1
1.3 User Prompt 2 2
1.4 TA Response 2 (Indexing) 2
1.5 User Prompt 3 3
1.6 TA Response 3 (Solution to Q1) 3
1.7 User Prompt 4 4
1.8 TA Response 4 (Solution to Q3) 4
1.9 User Prompt 5 5
1.10 TA Response 5 (Solution to Q4a-c) 5
1.11 User Prompt 6 5
1.12 TA Response 6 (Solution to Q4d-e) 5
1.13 User Prompt 7 6
1.14 TA Response 7 (Solution to Q5) 6

1 Conversation Trace

1.1 User Prompt 1

Initial Prompt ↘

```
You are a Teaching Assistant for a graduate-level Deep Learning course.
Your role is to rigorously, completely, and correctly solve homework problems only when instructed.

I will upload a PDF of the homework assignment.
Your workflow should be:

- Parse and analyze the PDF.
- Identify and index all question numbers in order, without solving anything yet.
- Wait for my explicit command: \Solve question X". When this command is given, solve only question
- All solutions must be written in clean, exportable LaTeX.
- After producing each solution, verify correctness, check for hallucinations, and correct any weak
- Maintain an internal record of the entire conversation trace, including all my prompts and your ou

Do you understand?
Please answer yes or no, and ask any clarifying questions if anything is ambiguous.
```

1.2 TA Response 1

```
Yes. I understand my role as a Teaching Assistant for a graduate-level Deep
Learning course and the specified workflow for handling the homework
assignment.

I will:
* Parse and analyze the uploaded PDF.
* Identify and index all question numbers in order.
```

* Wait for the explicit command "Solve question X" before providing a solution.
* Write all solutions in clean, exportable \LaTeX.
* Verify the correctness of each solution and reasoning.
* Maintain a record of the entire conversation trace.
* Provide the full trace in \LaTeX when requested.

I have no clarifying questions at this time. I am ready for you to upload the homework PDF.

1.3 User Prompt 2

Parsing of HW PDF

Here is the pdf of the homework

1.4 TA Response 2 (Indexing)

I have parsed and analyzed the uploaded PDF, "182_hw3.pdf".

Homework Question Index

1. Maximal Update Parameterization
   * (a) Scaling of expected squared RMS norm of $y=Wx$ at initialization.
   * (b) Scaling of expected RMS norm squared of unscaled feature change during training.
   * (c) Why the update rule only depends on d_1.
2. Visualizing Maximal Update Parameterization (Coding Question)
3. Maximal Update Parameterization Research
   * (a) Analysis of Figure 1 from the paper.
   * (b) Analysis of Table 3 from the paper.
   * (c) Analysis of Desideratum 1 and Condition 1 from a research paper.
4. Policy Gradient and the Reparameterization Gradient Estimator
   * (a) Express $\nabla_{\mu}\mathcal{F}(\mu)$ as an expectation for a Gaussian distribution (Poli
   * (b) Rewrite $\mathcal{F}(\mu)$ using an expectation over a standard normal distribution.
   * (c) Express $\nabla_{\mu}\mathcal{F}(\mu)$ using an expectation over $p_{\mu=0}(x)$ (Reparamet
   * (d) General Policy Gradient Estimator.
   * (e) General Reparameterization Gradient Estimator (Proof and special case).
5. Tensor Rematerialization
   * (a) Number of forward operations in a single backward pass.
   * (b) Number of loadmem operations in a single backward pass.
   * (c) Required speed of loaddisk operation for equal time.
6. Homework Process and Study Group
   * (a) Sources used.
   * (b) Collaborators.
   * (c) Time worked.

I am now waiting for your explicit command to solve a specific question, such as "Solve question 1(a)".

1.5 User Prompt 3

`Solve question 1`

1.6 TA Response 3 (Solution to Q1)

(a) Scaling of expected squared RMS norm of $y = Wx$ at initialization.

The squared RMS norm of $y \in \mathbb{R}^{d_2}$ is $\text{RMS}^2(y) = \frac{1}{d_2}\sum_{i=1}^{d_2} y_i^2$. For $y_i = \sum_{j=1}^{d_1} W_{ij}x_j$, where $W_{ij} \sim \mathcal{N}(0,1)$ and $x_j \sim \mathcal{N}(0,1)$ are i.i.d., the mean and variance of y_i are:

$$\mathbb{E}[y_i] = \sum_{j=1}^{d_1} \mathbb{E}[W_{ij}]\mathbb{E}[x_j] = 0$$

$$\text{Var}(y_i) = \mathbb{E}[y_i^2] = \sum_{j=1}^{d_1} \mathbb{E}[(W_{ij}x_j)^2] = \sum_{j=1}^{d_1} \mathbb{E}[W_{ij}^2]\mathbb{E}[x_j^2] = \sum_{j=1}^{d_1}(1)(1) = d_1$$

completely correct ✓

The expected squared RMS norm is:

$$\mathbb{E}[\text{RMS}^2(y)] = \mathbb{E}\left[\frac{1}{d_2}\sum_{i=1}^{d_2} y_i^2\right] = \frac{1}{d_2}\sum_{i=1}^{d_2}\mathbb{E}[y_i^2] = \frac{1}{d_2}\sum_{i=1}^{d_2} d_1 = d_1$$

Results:

?: Doesn't mention Xavier init

- The expected squared RMS norm is $\mathbb{E}[\text{RMS}^2(y)] = d_1$.
- This scales **linearly with the input dimension** d_1 (fan-in) and is **independent of** d_2 (fan-out).
- To ensure $\mathbb{E}[\text{RMS}^2(cWx)] = 1$, we need $c^2 \cdot d_1 = 1$. The constant to multiply W by is $c = \frac{1}{\sqrt{d_1}}$.

completely correct ✓

(b) Scaling of expected RMS norm squared of unscaled feature change $y' = sign(x_i g_i^T)x_i$* during training.

The unscaled feature change is $y' = \Delta W x_i$, where $\Delta W_{kj} = sign(g_{ik}x_{ij})$. The k-th component is $y'_k = \sum_{j=1}^{d_1}\Delta W_{kj}x_{ij} = \sum_{j=1}^{d_1} sign(g_{ik}x_{ij})x_{ij}$. The expected squared norm of y'_k is:

$$\mathbb{E}[(y'_k)^2] = \mathbb{E}\left[\sum_{j=1}^{d_1}\sum_{l=1}^{d_1} sign(g_{ik}x_{ij})x_{ij}\,sign(g_{ik}x_{il})x_{il}\right]$$

Diagonal terms ($j = l$)**:**

$$\sum_{j=1}^{d_1}\mathbb{E}[(sign(g_{ik}x_{ij})x_{ij})^2] = \sum_{j=1}^{d_1}\mathbb{E}[x_{ij}^2] = d_1 \quad (\text{since } sign^2(\cdot) = 1 \text{ and } \mathbb{E}[x_{ij}^2] = 1)$$

Off-diagonal terms ($j \neq l$)**:** The expectation factors due to independence of x_{ij} and x_{il} (conditional on g_{ik}):

$$\mathbb{E}_{x_{ij},x_{il}}[(sign(g_{ik}x_{ij})x_{ij})(sign(g_{ik}x_{il})x_{il}) \mid g_{ik}] = \left(\mathbb{E}_{x_{ij}}[sign(g_{ik}x_{ij})x_{ij} \mid g_{ik}]\right)^2$$

Since $sign(g_{ik}x_{ij})x_{ij} = sign(g_{ik})|x_{ij}|$, and $\mathbb{E}[|x_{ij}|] = \sqrt{2/\pi}$:

$$\mathbb{E}[(y'_k)^2] = d_1 + d_1(d_1-1)\cdot\mathbb{E}_{g_{ik}}\left[\left(sign(g_{ik})\sqrt{2/\pi}\right)^2\right] = d_1 + d_1(d_1-1)\frac{2}{\pi}$$

The expected squared RMS norm is $\mathbb{E}[\text{RMS}^2(y')] = \frac{1}{d_2}\sum_{k=1}^{d_2}\mathbb{E}[(y'_k)^2] = \mathbb{E}[(y'_1)^2]$:

$$\mathbb{E}[\text{RMS}^2(y')] = d_1\left(1 + (d_1-1)\frac{2}{\pi}\right)$$

Results:

- The expected RMS norm squared of the change in features scales **quadratically with** d_1 for large d_1 (approximately $\frac{2}{\pi}d_1^2$).
- It is **independent of** d_2.

✓: derivation is correct and Gemini went beyond the posted solutions (by analyzing off-diagonal entries)

✗: minor mistake: didn't include the multiplicative factor of $\frac{1}{d_1}$ in the final solution

(c) Why the update rule only depends on d_1.

✓: completely correct

The total expected squared ℓ_2 norm of the feature change is $\mathbb{E}[\|y'\|_2^2] = d_2 \cdot \mathbb{E}[(y'_1)^2]$. The RMS norm squared is defined with a scaling factor of $1/d_2$:

$$\mathbb{E}[\text{RMS}^2(y')] = \frac{1}{d_2}\mathbb{E}[\|y'\|_2^2] = \frac{1}{d_2}\left(d_2 \cdot \mathbb{E}[(y'_1)^2]\right) = \mathbb{E}[(y'_1)^2]$$

Since $\mathbb{E}[(y'_1)^2]$ only depends on the summation index limit d_1 (the fan-in), the overall expected squared RMS norm of the feature change is **independent of** d_2 (the fan-out). The definition of the RMS norm is specifically chosen to normalize away the dependence on the feature dimension d_2.

1.7 User Prompt 4

`Solve question 3. Below, I will attach the "Tensor Programs V" paper and the "Spectral Condition for Feature Learning" papers.`

1.8 TA Response 4 (Solution to Q3)

X: Hallucinated: Gemini summarized but didn't answer the question to part (a).

(a) Analysis of Figure 1 from "Tensor Programs V"

- **Left Figure (Standard Parameterization - SP):** The optimal learning rate for minimizing training loss **shifts significantly** (decreases) as the model width increases. This instability prevents zero-shot hyperparameter transfer.
- **Right Figure (Maximal Update Parameterization - μP):** The optimal learning rate for minimizing training loss **remains stable** (or invariant) as the model width increases.

The main observation is that μP stabilizes the optimal learning rate across different model widths, enabling **zero-shot hyperparameter transfer (μTransfer)** from small to large models.

(b) Analysis of Table 3 from "Tensor Programs V"

Under **Standard Parameterization (SP)**, the required Adam LR for hidden layers is $\eta_{\text{SP}} = \Theta(1/\text{fan}_{\text{in}})$. This scaling leads to feature updates Δh_ℓ whose RMS norm **vanishes** in the infinite-width limit ($\text{RMS}(\Delta h_\ell) \propto 1/\sqrt{\text{fan}_{\text{in}}} \to 0$ as $\text{fan}_{\text{in}} \to \infty$). This forces the network into the **Neural Tangent Kernel (NTK)** or **lazy training regime**, where features are frozen and **feature learning** is prevented. The μP scheme fixes this by ensuring $\eta_{\mu\text{P}} = \Theta(1)$.

✓: Correct.

(c) Analysis of Desideratum 1 and Condition 1 from "A Spectral Condition for Feature Learning"

✓

1. **Why Desideratum 1 is desired for Feature Learning. Desideratum 1** requires that both the feature norm $||h_\ell(x)||_2$ and the feature update norm $||\Delta h_\ell(x)||_2$ must be $\Theta(\sqrt{n_\ell})$, which is equivalent to requiring the **Root Mean Square (RMS) norm** of both to be $\boldsymbol{\Theta(1)}$. The requirement $\text{RMS}(\Delta h_\ell) = \Theta(1)$ ensures that the feature vector changes by a **non-vanishing** amount relative to the element size in the large-width limit, which is the definition of **feature learning**.
2. **How Condition 1 implies $||\Delta h_\ell(x)||_2 \leq \Theta(\sqrt{n_\ell})$.** The feature update norm satisfies the inequality:

X: made wrong assumption, leading to hallucination.

$$||\Delta h_\ell(x)||_2 \leq ||\Delta W_\ell||_{\text{spec}} \cdot ||h_{\ell-1}(x)||_2$$

Condition 1 specifies $||\Delta W_\ell||_{\text{spec}} = \Theta(1)$. **Desideratum 1** requires the input feature norm $||h_{\ell-1}(x)||_2 = \Theta(\sqrt{n_{\ell-1}})$. Substituting these gives the upper bound:

$$||\Delta h_\ell(x)||_2 \leq \Theta(1) \cdot \Theta(\sqrt{n_{\ell-1}}) = \boldsymbol{\Theta}(\sqrt{\mathbf{n}_\ell}) \quad \underline{\text{(assuming } n_\ell \approx n_{\ell-1})}$$

✓

3. **Key assumption for getting the lower bounds.** The key assumption required to achieve the lower bound $||\boldsymbol{\Delta}\mathbf{h}_\ell(\mathbf{x})||_\mathbf{2} = \boldsymbol{\Omega}(\sqrt{\mathbf{n}_\ell})$ (saturating the Θ requirement) is **Spectral Alignment**: The input feature vector $h_{\ell-1}$ must be sufficiently aligned with the principal singular vector of the weight update matrix ΔW_ℓ.

1.9 User Prompt 5

`Solve question 4 parts (a), (b), and (c)`

1.10 TA Response 5 (Solution to Q4a-c)

The objective function is $\mathcal{F}(\mu) = \mathbb{E}_{x\sim p_\mu(x)}[f(x)]$, where $p_\mu(x) = \mathcal{N}(x;\mu,\Sigma)$ and Σ is fixed.

(a) Express $\nabla_\mu \mathcal{F}(\mu)$ as an expectation for a Gaussian distribution (Policy Gradient Estimator).

Using the Policy Gradient Estimator (Log-Derivative Trick):

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{x\sim p_\mu(x)}\left[f(x)\nabla_\mu \log p_\mu(x)\right]$$

✓: Correct

For $p_\mu(x) = \mathcal{N}(x;\mu,\Sigma)$, the score function is $\nabla_\mu \log p_\mu(x) = \Sigma^{-1}(x-\mu)$.

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)}\left[f(x)\Sigma^{-1}(x-\mu)\right]$$

(b) Rewrite $\mathcal{F}(\mu)$ using an expectation over a standard normal distribution.

Using the Reparameterization Trick, let $x = g(\mu,z) = \mu + Lz$, where $z\sim\mathcal{N}(0,I)$ and $\Sigma = LL^T$.

$$\mathcal{F}(\mu) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[f(\mu+Lz)]$$

✓: Correct

(c) Express $\nabla_\mu \mathcal{F}(\mu)$ using an expectation over $p_{\mu=0}(z) = \mathcal{N}(0,\Sigma)$ (Reparameterization Estimator).

Using the reparameterization $x = \mu + z$, with $z\sim\mathcal{N}(0,\Sigma)$:

$$\nabla_\mu \mathcal{F}(\mu) = \nabla_\mu \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}[f(\mu+z)] = \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}[\nabla_\mu f(\mu+z)]$$

By the chain rule, $\nabla_\mu f(\mu+z) = \nabla_x f(x)|_{x=\mu+z}$.

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}[\nabla_x f(\mu+z)]$$

✓: Correct

1.11 User Prompt 6

`Solve question 4 parts (d), (e)`

1.12 TA Response 6 (Solution to Q4d-e)

(d) General Policy Gradient Estimator (PG).

✓: Correct, but skipped derivation.

The formula is:

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{x\sim p_\mu(x)}\left[f(x)\nabla_\mu \log p_\mu(x)\right]$$

Advantage: Generality (Applicable to any differentiable distribution, including **discrete** distributions). **Disadvantage: High Variance** (The product form leads to high variance samples).

(e) General Reparameterization Gradient Estimator (RG).

1. Proof of the General Estimator: For $x = g(\mu,z)$, $z\sim q(z)$ independent of μ:

$$\nabla_\mu \mathcal{F}(\mu) = \nabla_\mu \mathbb{E}_{z\sim q(z)}[f(g(\mu,z))] = \mathbb{E}_{z\sim q(z)}[\nabla_\mu f(g(\mu,z))]$$

Using the chain rule, $\nabla_\mu f(g(\mu,z)) = \nabla_x f(g(\mu,z))\cdot\nabla_\mu g(\mu,z)$:

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{z\sim q(z)}\left[\nabla_x f(g(\mu,z))\cdot\nabla_\mu g(\mu,z)\right]$$

✓: Correct.

2. Special Case $p_\mu(x) = \mathcal{N}(x;\mu,I)$: Here $g(\mu,z) = \mu + z$ and $\nabla_\mu g(\mu,z) = I$.

$$\nabla_\mu \mathcal{F}(\mu) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\nabla_x f(\mu+z)\right]$$

1.13 User Prompt 7

`Solve question 5`

1.14 TA Response 7 (Solution to Q5)

Let L be the number of layers, and n be the operations/memory per layer. $t_{\text{fwd}} = 20$ ns.

(a) Number of forward operations in a single backward pass (Tensor Rematerialization).

The total forward recomputations $N_{\text{fwd}}^{\text{bwd}}$ is the sum of operations to recompute A_{l-1} from A_0 for layers $l = 1$ to L:

$$N_{\text{fwd}}^{\text{bwd}} \approx \sum_{l=1}^{L} \sum_{j=1}^{l-1} n = n \sum_{j=0}^{L-1} j = n \frac{L(L-1)}{2}$$

✓: Correct

(b) Number of loadmem operations in a single backward pass.

In a fully rematerialized scheme, activations $A_1, \ldots, A_L$ are recomputed, not loaded from fast memory (DRAM).

$$N_{\text{loadmem}} = 0$$

✗: Incorrect. Made bad assumption.

(c) Required speed of loaddisk operation for equal time.

We equate the time for Tensor Rematerialization (Time_{TR}) and Disk Loading (Time_{DL}). Let $\text{Time}_{\text{bwd}}^{\text{std}} = Ln \cdot t_{\text{fwd}}$ be the standard backward computation time. $M = Ln$ is the total memory.

$$\text{Time}_{\text{TR}} = \text{Time}_{\text{bwd}}^{\text{std}} + N_{\text{fwd}}^{\text{bwd}} \cdot t_{\text{fwd}}$$

$$\text{Time}_{\text{DL}} = \text{Time}_{\text{bwd}}^{\text{std}} + M \cdot t_{\text{disk}}$$

Setting $\text{Time}_{\text{TR}} = \text{Time}_{\text{DL}}$:

$$N_{\text{fwd}}^{\text{bwd}} \cdot t_{\text{fwd}} = M \cdot t_{\text{disk}}$$

Substituting the values:

$$\left(n \frac{L(L-1)}{2}\right) \cdot t_{\text{fwd}} = (Ln) \cdot t_{\text{disk}}$$

Solving for t_{disk}:

$$t_{\text{disk}} = \frac{L(L-1)}{2L} \cdot t_{\text{fwd}} = \frac{L-1}{2} \cdot t_{\text{fwd}}$$

With $t_{\text{fwd}} = 20$ ns:

$$t_{\text{disk}} = \frac{L-1}{2} \cdot 20 \text{ ns} = (L-1) \cdot 10 \text{ ns}$$

✗: Incorrect by extension from Part (b). Otherwise, logic and formulas are correct.

The required speed of the loaddisk operation is $(\mathbf{L-1}) \cdot \mathbf{10}$ ns.