STAT 946 Fall 2025 | Scribes: Daniel Zhang
October 29, 2025 | Instructor: Mufan Li

STAT 946 - Topics in Probability and Statistics: Mathematical Foundations of Deep Learning Lecture 13: Feature Learning

1 Feature Learning (Yang & Hu, 2021)

Consider an L-layer fully-connected network parameterized in the *maximal update parametrization* (also called μP). The model is defined as:

$$f(x;\theta) = \boxed{\frac{1}{n}} \underset{(1\times n)}{W_d}\, \varphi(h_d),$$

where the hidden representations satisfy

$$h_{l+1} = \frac{1}{\sqrt{n}} \underset{(n\times n)}{W_l}\, \varphi(h_l), \quad l = 1, \dots, d-1$$

with input layer

$$h_1 = \frac{1}{\sqrt{n_0}} \underset{(n\times n_0)}{x}.$$

The learning rate scales with width:

$$\boxed{\eta = \eta_0\, n}.$$

This scaling is known as the **Maximal Update Parametrization** (μP), which ensures stable feature learning as width $n \to \infty$.

Definition:

- Δh_l: the change in the hidden representation h_l after one gradient step.
- We say that layer l *learns features* if $\Delta h_l = \Theta(1)$ (i.e., the change is order-one in width).

Remark:

$$\Delta h_l \;\simeq\; \frac{1}{\sqrt{n}}\, \Delta W_l\, \varphi(h_l) \;+\; \frac{1}{\sqrt{n}}\, W_l\, \Delta\varphi(h_l).$$

- If we **only train** the input weights W_0, then $\Delta h_1 = \Theta(1)$.
- If we **freeze** all deeper layers (i.e., $\Delta W_l = 0$ for $l \geq 1$),

$$\implies \Delta h_l = \Theta(1), \quad \forall\, l.$$

Definition: A parameterization (weight prefactors, learning rates, etc.) is said to satisfy maximal update if

$$\frac{1}{\sqrt{n}} \Delta W_l \, \varphi(h_l) = \Theta(1), \quad \forall\, l.$$

Remark: If $n \to \infty$ is the only limit, then μP is the unique parameterization where all W_l contributes to feature learning

Recall:

$$[\theta_1, \theta_2, \ldots, \theta_n] \,/ \overset{\text{permutation invariant}}{\sim} \implies \frac{1}{n}\sum_{i=1}^{n} \delta_{\theta_i} \implies \text{mean-field network.}$$

Consider $d = 2$, linear

$$f(x;\theta) = \frac{1}{n} W_2 \frac{1}{\sqrt{n}} W_1 \frac{1}{\sqrt{n_0}} W_0 x$$

$$= \frac{1}{n} \sum_{\boxed{i=1}}^{n} W_{2,i} \frac{1}{\sqrt{n}} \sum_{\boxed{j=1}}^{n} \boxed{W_{1,ij}} \frac{1}{\sqrt{n_0}} \langle W_{0,j}, x \rangle.$$

permutation invariant

permutation invariant

$W_1 \in \mathbb{R}^{n \times n}$ is a separately exchangeable array (when W_1, W_0 are not trained and i.i.d.).

$$\big[\, W_{1,ij}(t) \,\big]_{i,j} \overset{d}{=} \big[\, W_{1,\sigma(i)\,\sigma'(j)}(t) \,\big]_{i,j}, \qquad \sigma, \sigma' \in S_n.$$

Does not admit an explicit quotient representation.

Aldous–Hoover Representation. There exists a measurable function

$$f : [0,1]^4 \to \mathbb{R} \quad \text{such that} \quad \big[\, W_{1,ij} \,\big]_{i,j} \overset{d}{=} \big[\, f(U, U_i, V_j, U_{ij}) \,\big]_{i,j},$$

where

$$U,\, U_i,\, V_j,\, U_{ij} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}([0,1]).$$

<u>Remark.</u> No mean-field PDEs exist for deep networks under μP.

State Space

$$\begin{aligned} \text{Mean Field (MF):} \quad & \theta \text{ is full state} && \Longrightarrow \text{ no finite representation,} \\ \text{NTK:} \quad & f \text{ outputs} && \Longrightarrow \text{ poor feature learning model,} \\ \text{DMFT:} \quad & \big[f^\alpha, h^\alpha, z^\alpha\big]_{\alpha=1}^m && \Longrightarrow \text{ depends on } m = \#\text{data.} \end{aligned}$$

2 History and Refs

$$\left.\begin{array}{r}\text{Statistical Physics - Martin, Siggia, Rose (1973)} \\ \text{Recurrent NN - Sompolinsky, Crisanti, Sommers (1988)}\end{array}\right\} \text{Not rigorous}$$

Spin Glass - Ben-Arous, Guionnet (1995)
High-Dimensional SGD – Celentano, Cheng, Montanari (2021)
NN ($n \to \infty$):

- Yang and Hu (2021)
- Bordelon and Pehlevan (2022) $\rightarrow$ Not rigorous

3 Dynamical MF Theory for Training Dynamics

<u>Recall</u>:

$$\boxed{h^\alpha_{l+1}} = \frac{1}{\sqrt{n}} W_l\, \varphi(h^\alpha_l), \qquad \text{(post–activation)}$$

$$g^\alpha_l = \sqrt{n}\, \frac{\partial f}{\partial h^\alpha_l},$$

$$\boxed{z^\alpha_l} = \frac{1}{\sqrt{n}} W_l^\mathsf{T} g^\alpha_{l+1} = \frac{1}{\sqrt{n}} W_l^\mathsf{T} \operatorname{diag}\!\big(\varphi'(h^\alpha_{l+1})\big)\, z^\alpha_{l+1}, \qquad \text{(pre–activation)}$$

$h^\alpha_{l+1},\ z^\alpha_l$ can be viewed as states

Markov View.

$$\theta(k) \xrightarrow{\text{GD}} \theta(k+1)$$

$$h(k) \xrightarrow{\text{GD}} h(k+1)$$

$$z(k) \xrightarrow{\text{GD}} z(k+1)$$

However, note that the weights $\theta(k+1)$ rely on information from $\theta(k)$ (excluding h, z).

$$\theta(k) \xrightarrow{n\to\infty} \text{depends only on } \{\, h^\alpha(t),\; z^\alpha(t)\,\}_{t\le k,\; \alpha\in[m]}.$$

<u>**Good:**</u> Only neuron-wise states needed as $n \to \infty$.
<u>**Bad:**</u> Requires full history $(t \le k) \;\Rightarrow\; non-Markovian.$

<u>**Full DMFT Equations**</u>

Time derivative:

$$\delta_t \theta(t) = -\,\eta\, \nabla_\theta L(\theta(t)), \qquad \eta = n.$$

The equations:

$$h_l^\alpha(t) \;=\; u_l^\alpha(t) \;+\; \int_0^t ds \sum_{\beta=1}^{n} \Big[A_l^{\alpha\beta}(t,s) \;+\; \Delta^\beta(s)\, \Phi_{l-1}^{\alpha\beta}(t,s) \Big]\, z_l^\alpha(s)\, \varphi'\big(h_l^\alpha(s)\big).$$

where:

- h_l^α is a single neuron state (scalar, i-th index),
- $u_l^\alpha(t) \sim \mathrm{GP}\Big(0,\; [\,\Phi_{l-1}^{\alpha\beta}(t,s)\,]_{\alpha\beta}\Big)$,
- $A_l^{\alpha\beta}(t,s) = \frac{1}{n}\sum_{j=1}^{n} \frac{\partial\, \varphi\left(h_l^\alpha(t)\right)_j}{\partial\, r_l^\beta(s)_j}$,
- $\Delta^\beta(s) = -\,\frac{\partial L}{\partial f^\beta} = y^\beta - f^\beta$ (for square loss),
- $\Phi_{l-1}^{\alpha\beta}(t,s) = \lim_{n\to\infty} \frac{1}{n} \left\langle \varphi\big(h_{l-1}^\alpha(t)\big),\; \varphi\big(h_{l-1}^\beta(s)\big) \right\rangle$,

- $z_l^\alpha(s)$ is a single-neuron (scalar, i-th index) adjoint / backward state.

$$z_l^\alpha(t) \;=\; r_l^\alpha(t) \;+\; \int_0^t ds \sum_{\beta=1}^{n} \Big[B_l^{\alpha\beta}(t,s) \;+\; \Delta^\beta(s)\, G_{l+1}^{\alpha\beta}(t,s) \Big]\; \varphi\big(h_l^\alpha(s)\big).$$

where:

- z_l^α is a single-neuron (scalar, i–th index) backward state,
- $r_l^\alpha(t) \sim \mathrm{GP}\Big(0,\; [\,G_{l-1}^{\alpha\beta}(t,s)\,]_{\alpha\beta}\Big)$,
- $B_l^{\alpha\beta}(t,s) = \dfrac{1}{n}\sum_{i=1}^n \dfrac{\partial\, g_{l+1}^\alpha(t)_i}{\partial\, u_l^\beta(s)_i}$,
- $G_{l+1}^{\alpha\beta}(t,s) = \lim\limits_{n\to\infty} \dfrac{1}{n}\big\langle g_{l+1}^\alpha(t),\; g_{l+1}^\beta(s)\big\rangle$, (average over neuron index)

  $\Leftrightarrow \displaystyle\int \cdots d\rho, \qquad d\rho = \frac{1}{n}\sum_{i=1}^{n} \delta_{(h_{l,i},\, z_{l,i})}.$

<u>**Remarks:**</u> Mean Field (particle) equations

- RHS (dynamics) only depends on (h,z) and history $\implies$ the system is <u>closed</u>
- u, r, A, B (depend on history) contributed by weights

Gaussian conditioning:

$$W \,\big|\, (W\varphi,\; g^\mathsf{T} W) \;\stackrel{d}{=}\; (*)$$

$$(*) \;=\; \underbrace{P_g\, W \;+\; W\, P_\varphi \;-\; P_g\, W\, P_\varphi}_{\substack{g,\,\varphi \text{ contain history} \\ \implies A,B \text{ kernels}}} \;+\; \underbrace{P_g^\perp\; \tilde{W}\; P_\varphi^\perp}_{\text{Gaussian process (GP)}} .$$

where:

- $P_g^\perp$ corresponds to the backward kernel G,
- $\tilde{W}$ is an independent copy of W,

- $P_\varphi^\perp$ corresponds to the forward kernel Φ.

$$h_l(t) \;=\; \text{GP} \;+\; \underbrace{\int_0^t ds\;\Theta(1)}_{\substack{\text{feature learning / maximal update} \\ \Delta W_{l-1}\,\varphi(h_{l-1})}}$$

4 Heuristic Derivation for Scaling (Not DMFT)

- For simplicity: $m = 1$ (single data point), $\varphi(x) = x$ (linear).
- $f = \dfrac{1}{\gamma\sqrt{n}}\,W_d\,h_d$
- $h_{l+1} = \dfrac{1}{\sqrt{n}}\,W_l\,h_l$
- $h_1 = W_0\,X$
- $\partial_t\theta = -\,\eta\,\nabla_\theta L(\theta)$

Goal: Set γ, η as functions of n.
Recall: $a_n \sim b_n$ if $\dfrac{a_n}{b_n} \to$ const.
i.e. $a_n = \Theta(n^p) \iff b_n = \Theta(n^p)$.

$$W_l(t) = W_l(0) \;+\; \frac{\eta}{\gamma n}\int_0^t ds\;\Delta(s)\,z_{l+1}(s)\,h_l(s)^\mathsf{T}.$$

$$h_{l+1}(t) = \underbrace{\frac{1}{\sqrt{n}}\,W_l(0)\,h_l(t)}_{\substack{\text{CLT scaling} \\ \text{(GP term)}}}$$

$$+\;\frac{1}{\sqrt{n}}\cdot\frac{\eta}{\gamma n}\int_0^t ds\;\Delta(s)\,z_{l+1}(s)\;\underbrace{h_l(s)^\mathsf{T}h_l(s)}_{\substack{\frac{1}{n}h^\mathsf{T}h=\Phi \\ \text{(kernel term)}}}.$$

$$\sim\;\text{GP}\;+\;\underbrace{\frac{\eta}{\gamma\sqrt{n}}}_{\sim 1}\int_0^t ds.$$

$$h_{d+1} = \frac{1}{\sqrt{n}} W_d \, h_d, \qquad f = \frac{1}{\gamma} \, h_{d+1}.$$

$$K \;=\; \big\langle \nabla_\theta h_{d+1}, \, \nabla_\theta h_{d+1} \big\rangle \;\sim\; 1.$$

$$\big\langle \nabla_\theta f, \, \nabla_\theta f \big\rangle \;=\; \frac{1}{\gamma^2} \, K \;\sim\; \frac{1}{\gamma^2}.$$

$$\partial_t f = - \, \eta \, \big\langle \nabla_\theta f, \, \nabla_\theta f \big\rangle \, \Delta = \underbrace{\frac{\eta}{\gamma^2}}_{\sim 1} \, \underbrace{K \, \Delta}_{\sim 1}.$$

$$\Longrightarrow \; \eta \;\sim\; \gamma^2$$

$$\frac{n}{\gamma} \;\sim\; \gamma \qquad \Longrightarrow \qquad \boxed{\gamma \;\sim\; \sqrt{n} \quad \Longrightarrow \quad \eta \;\sim\; n} \quad \Rightarrow \quad \mu\mathrm{P}.$$