ECE 57000

Special topics: modern ML theory/concepts

The benefits of Over-parameterization

Chaoyue Liu

Fall 2024

Content

What is over-parameterization

Benefits of over-parameterization
- Double descent
- Automatic variance reduction
- Transition to linearity

Over-parameterization

The number of model parameters p is greater than the number of training samples n

The goal of ML training is to fit the data (or at least *approximately* fit):

$$f_i(\mathbf{w}) = f(\mathbf{w}; \mathbf{x}_i) = y_i$$

Another perspective: solve the set of equations

$$\left.\begin{aligned} f_1(\mathbf{w}) &= y_1 \\ f_2(\mathbf{w}) &= y_2 \\ &\cdots \\ f_n(\mathbf{w}) &= y_n \end{aligned}\right\} n \text{ equations (constraints)}$$

p parameters (degrees of freedom)

In over-parameterization regime, the data can be exactly fit (namely, the set of equations can be exactly solved) -- **Interpolation**

Interpolation

the data can be exactly fit :

$$f_i(\mathbf{w}) = f(\mathbf{w}; \mathbf{x}_i) = y_i, \qquad \forall i \in [n]$$

Training loss can be exactly zero $\mathcal{L}(\mathbf{w}^*) = 0$.

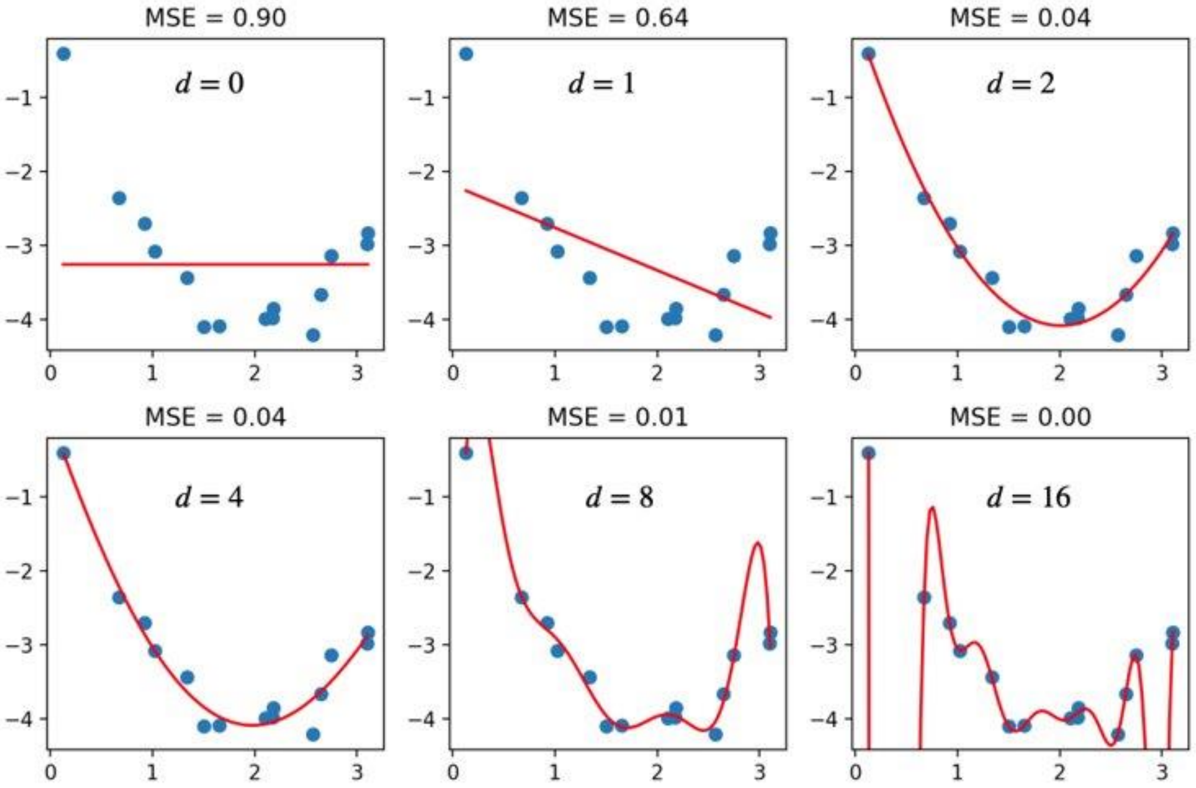
Moreover, each individual loss can be exactly zero:

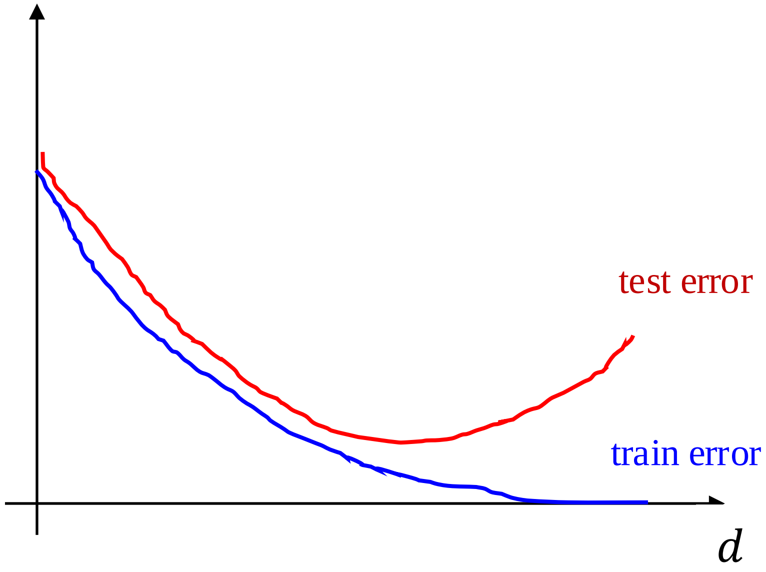
$$\ell_i(\mathbf{w}^*) = 0, \qquad \forall i \in [n]$$

Overfitting?

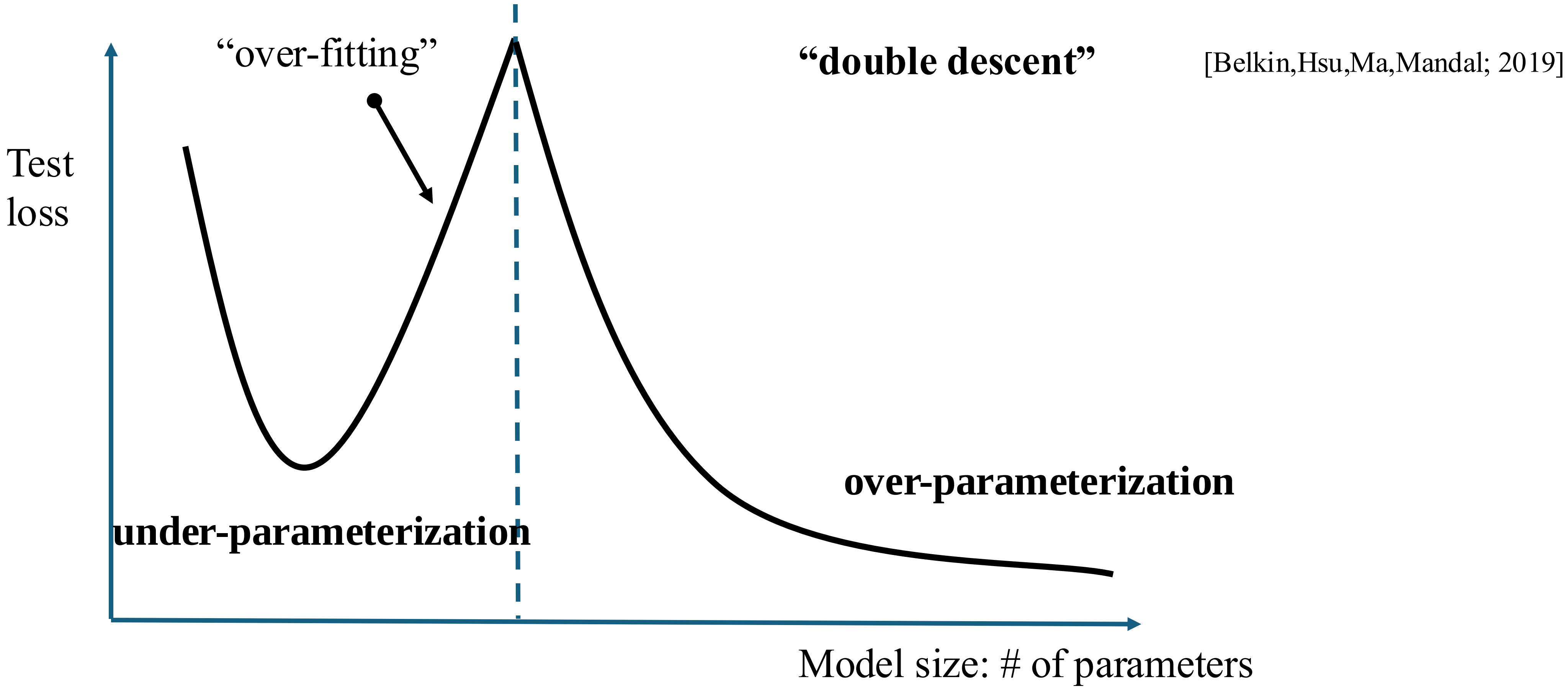
Fitting data with polynomial features:

- Number of training samples n = 16
- Number of model parameters p = d





Double descent



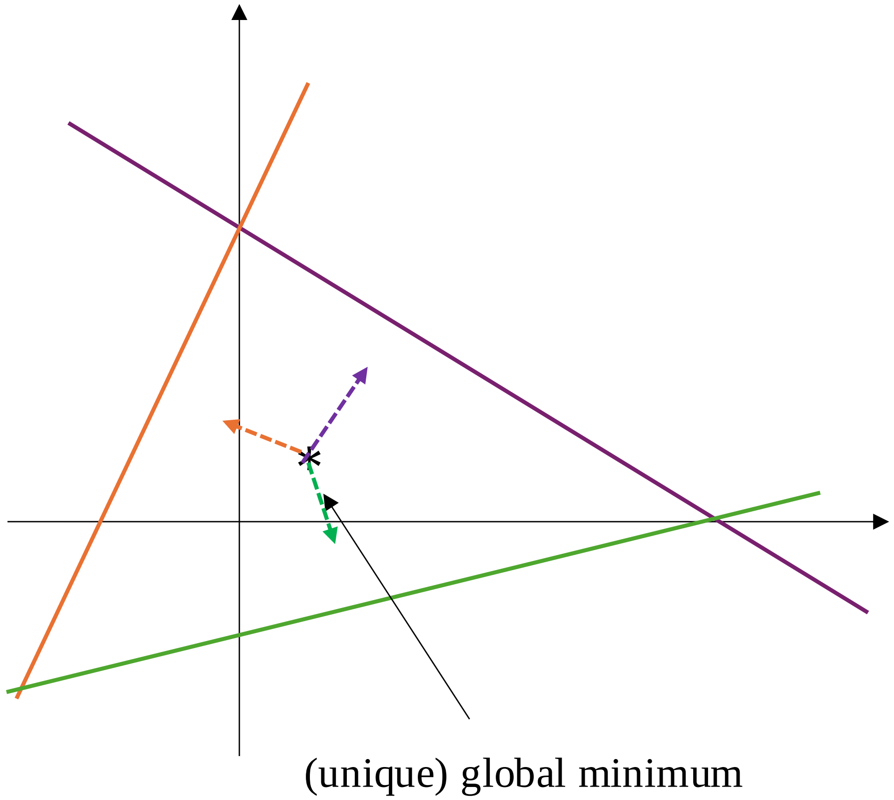
Automatic variance reduction

In **under-parameterized** regime,
SGD needs a **decreasing** learning rate to converge: $\eta_t \to 0$

Example:

Dataset:

$$\begin{aligned} \mathbf{x}_1 &= (1,2), & y_1 &= 4 & & & w_1 + 2 \cdot w_2 &= 4 \\ \mathbf{x}_2 &= (-2,1), & y_1 &= 2 & &\Longrightarrow & -2 \cdot w_1 + w_2 &= 2 \\ \mathbf{x}_3 &= (1,-4), & y_1 &= 4 & & & w_1 - 4 \cdot w_2 &= 4 \end{aligned}$$



Stochastic gradient $\nabla \ell_i(\mathbf{w}) = \nabla \mathcal{L}(\mathbf{w}) + \epsilon$
Stochastic noise ϵ: $Var[\epsilon] \neq 0$

Automatic variance reduction

In over-parameterized regime,
SGD with a **constant** learning rate can converge: $\eta_t = \eta$

Intuition:

- **At global minima $\mathbf{w}^*$** where $\mathcal{L}(\mathbf{w}^*) = 0$, each $\ell_i(\mathbf{w}^*) = 0$, because $f_i(\mathbf{w}^*) = y_i$. Namely, stochastic noise $\epsilon = \nabla\ell_i(\mathbf{w}^*) - \nabla\mathcal{L}(\mathbf{w}^*) = 0$
- At other points $\mathbf{w} \neq \mathbf{w}^*$, stochastic gradients $\nabla\ell_i = (f_i - y_i) \cdot \nabla f_i$, therefore, $Var[\epsilon] \sim \mathbb{E}[|f_i(\mathbf{w}) - y_i|^2]$; noise variance decreases as training loss decreases.

For more theory, see https://arxiv.org/pdf/1810.13395

Transition to linearity

A neural network f is a function $f(\theta, x)$

network f as a **function of input** $f_\theta(x)$

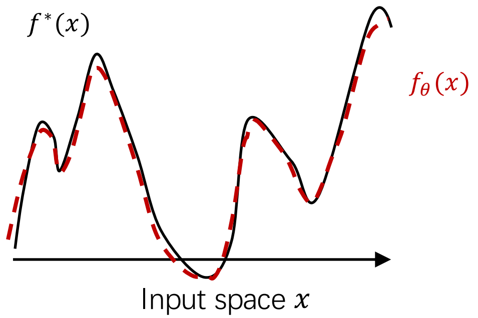
Universal approximation [Hornik et al. 1989]:

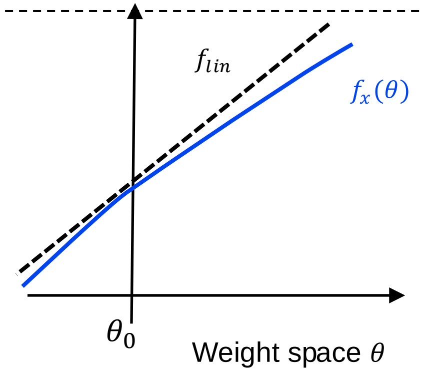
- Neural networks can approximate any *continuous* functions on a finite domain
- Larger network width $\Longrightarrow$ better approximation
- Infinite network width $\Longrightarrow$ exactly match the target function.

Fixing input x, network f as a **function of weights** $f_x(\theta)$

Transition to linearity [**Liu**, Zhu, Belkin NeurIPS 20]:

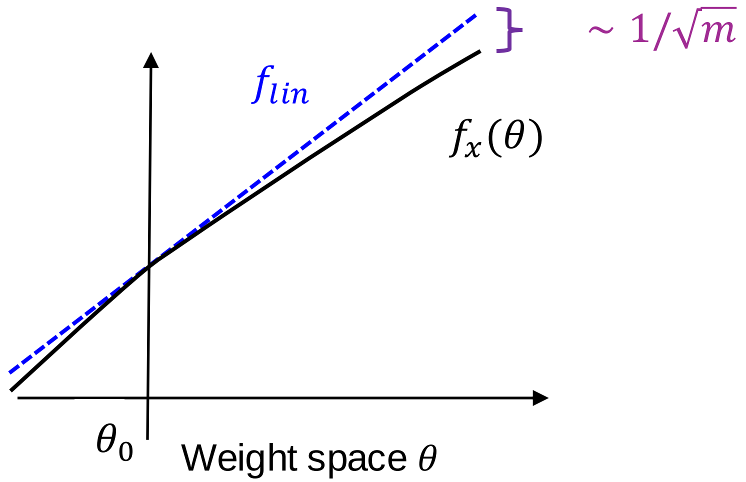
- Neural networks $f_x(\theta)$ is close to a linear function: non-linear terms are small, on a finite domain
- Larger network width $\Longrightarrow$ smaller non-linear term
- Infinite network width $\Longrightarrow$ non-linear term vanishes





Transition to linearity

$$f(\theta) = f(\theta_0) + \nabla f(\theta_0)(\theta - \theta_0) + \underbrace{\frac{1}{2}(\theta - \theta_0)^T H(\theta_0)(\theta - \theta_0) + \cdots}_{\text{Vanishes as network width } m \to \infty}$$



Transition to linearity

Why is *transition to linearity* useful?

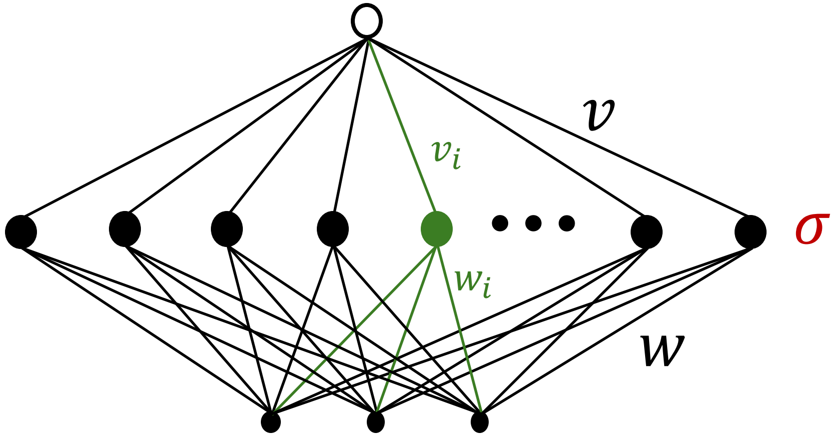
- Simplifies the neural network function (w.r.t. parameters θ)
- Simplifies loss landscape
- Theoretical guarantees for convergence of gradient descent algorithms (including SGD)

Illustration: two-layer network

Two-layer neural network (f):

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \sigma\left(w_i x\right)$$

Initialization: $v_{0,i} \sim \mathcal{N}(0,1)$; $w_{0,i} \sim \mathcal{N}(0,1)$.



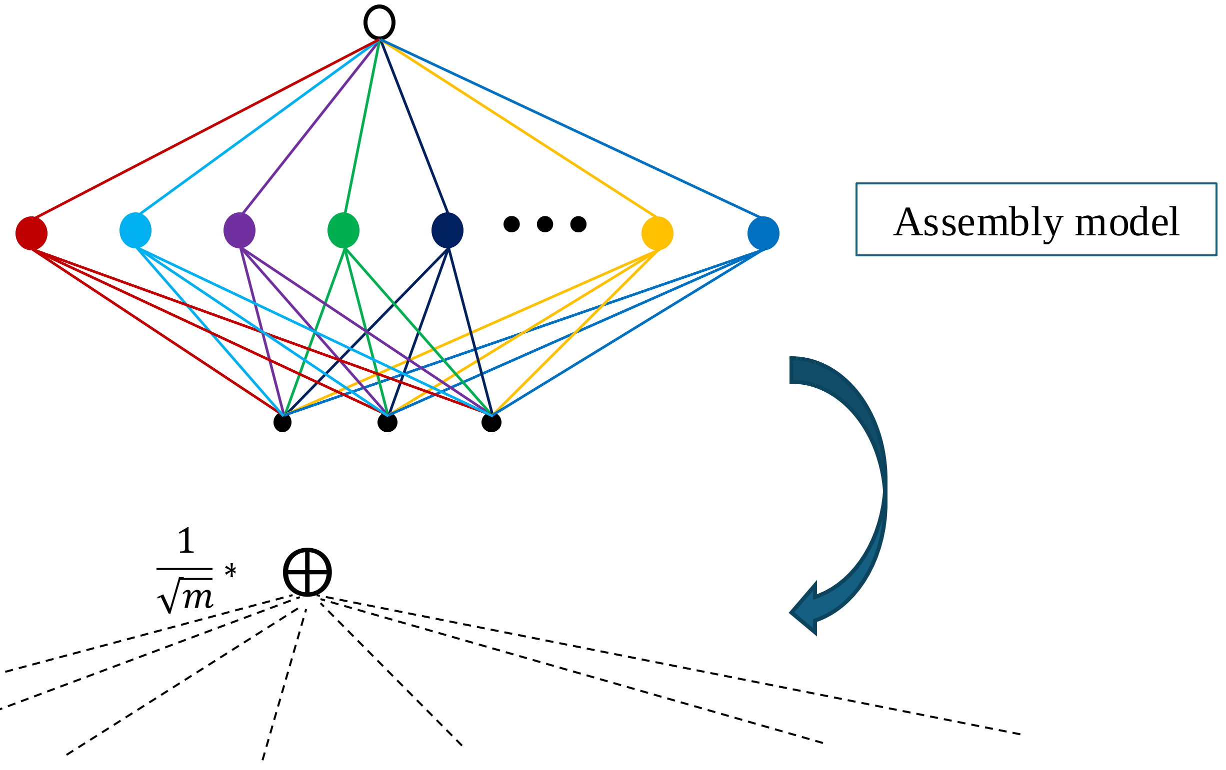
Notations

- weights $\theta = \{\theta_i\}_{i=1}^{m}$, $\theta_i = (v_i, w_i)$,
- Non-linear activation σ: e.g., *sigmoid*, *tanh*, ReLU

Assembly model view

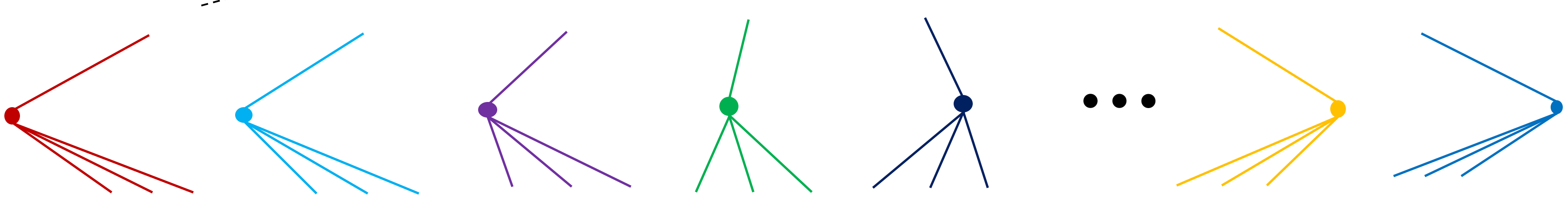
Network

$$f(\theta) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \sigma\,(w_i x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} g_i$$



Observation 1: Assembly operation is a scaled summation

Sub-models $g_i = v_i \sigma(w_i x)$



Observation 2: sub-model independence -- sub-models share no weights

Transferring structure to mathematical properties

Network $f(\theta) = \frac{1}{\sqrt{m}}\sum_{i=1}^{m} v_i\sigma(w_i x) \triangleq \frac{1}{\sqrt{m}}\sum_{i=1}^{m} g_i(\theta)$

Taylor expansion: (at initialization θ_0)

Lagrange remainder

$$f(\theta) = \underbrace{f(\theta_0) + \nabla f(\theta_0)(\theta - \theta_0)}_{\text{linear part: } f_{lin}(\theta)} + \underbrace{\frac{1}{2}(\theta - \theta_0)^T H(\xi)(\theta - \theta_0)}_{\text{non-linear part:} \mathcal{R}(\theta)}$$

*: ξ is between θ and θ_0

Hessian $H = \frac{\partial^2 f}{\partial \theta^2}$

Transferring structure to mathematical properties

Lagrange remainder (the non-linear part):

$$\mathcal{R}(\theta) = \frac{1}{2}(\theta - \theta_0)^T H_f(\xi)(\theta - \theta_0) \quad = \frac{1}{2\sqrt{m}} \sum_{i=1}^{m} (\theta - \theta_0)^T H_{g_i}(\xi)(\theta - \theta_0)$$

Obs 1: assembly

$$H_f(\xi) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} H_{g_i}(\xi)$$

Transferring structure to mathematical properties

Lagrange remainder (the non-linear part):

$$\mathcal{R}(\theta) = \frac{1}{2}(\theta - \theta_0)^T H_f(\xi)(\theta - \theta_0) \quad = \frac{1}{2\sqrt{m}} \sum_{i=1}^{m} (\theta - \theta_0)^T H_{g_i}(\xi)(\theta - \theta_0) \quad = \frac{1}{2\sqrt{m}} \sum_{i=1}^{m} \left(\theta_i - \theta_{0,i}\right)^T H_{g_i}(\xi)\left(\theta_i - \theta_{0,i}\right)$$

Obs 1: assembly

Obs 2: Independence

$$\left(\theta_j - \theta_{0,j}\right)^T H_{g_i}(\xi)\left(\theta_j - \theta_{0,j}\right) = 0$$

$$\forall j \neq i$$

Transferring structure to mathematical properties

Lagrange remainder (the non-linear part):

$$\mathcal{R}(\theta) = \frac{1}{2\sqrt{m}} \sum_{i=1}^{m} \left(\theta_i - \theta_{0,i}\right)^T H_{g_i}(\xi)\left(\theta_i - \theta_{0,i}\right)$$

each sub-model g_i is smooth:
$||H_{g_i}(\xi)||_{sp} \leq \beta$
β is a constant

$$|\mathcal{R}(\theta)| \leq \frac{1}{2\sqrt{m}} \sum_{i=1}^{m} ||H_{g_i}(\xi)||_{sp} \cdot ||\theta_i - \theta_{0,i}||^2$$

$$\leq \frac{\beta}{2\sqrt{m}} ||\theta - \theta_0||^2$$

$$\sim O\left(\frac{1}{\sqrt{m}}\right), \quad \text{for finite } ||\theta - \theta_0||^2$$

in a finite domain. e.g., a ball
$B(\theta_0, R)$ of finite radius R

Transition to Linearity

Lagrange remainder (the non-linear part):

$$|\mathcal{R}(\theta)| \sim O(\frac{1}{\sqrt{m}}), \quad \text{for finite } ||\theta - \theta_0||^2$$

When m is large, $|\mathcal{R}(\theta)|$ is small;
When $m \to \infty$, $|\mathcal{R}(\theta)| \to 0$.

Transition to linearity:

$$f(\theta) = \underbrace{f(\theta_0) + \nabla f(\theta_0)(\theta - \theta_0)}_{\text{Linear part: } f_{lin}(\theta)} + \underbrace{\cancel{\frac{1}{2}(\theta - \theta_0)^T H(\xi)(\theta - \theta_0)}}_{\text{Non-linear part: } \mathcal{R}(\theta)}$$

equivalently $\quad \|H(\theta)\|_{sp} \sim O\left(\frac{1}{\sqrt{m}}\right), \forall\theta \in B(\theta_0, R)$, for $m \to \infty$.

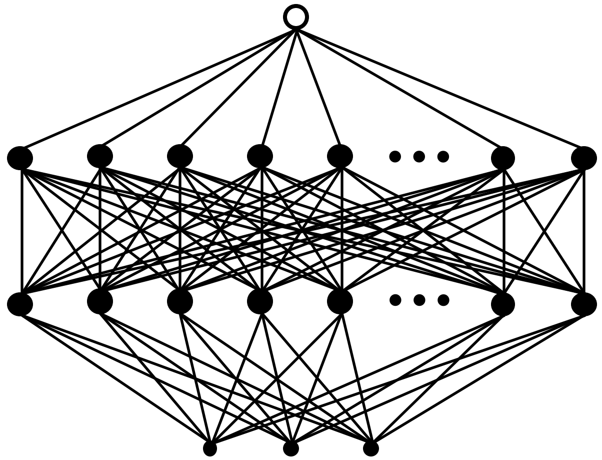
How about deep networks?

$$\alpha^{(0)} = x$$

$$\tilde{\alpha}^{(l)} = \frac{1}{\sqrt{m_{l-1}}} \sum_{i=1}^{m_{l-1}} w_i^{(l)} \alpha_i^{(l-1)}, \forall l = 1,2,\cdots,L$$

$$\alpha^{(l)} = \sigma\left(\tilde{\alpha}^{(l)}\right), \forall l = 1,2,\cdots,L-1$$

$$f = \tilde{\alpha}^{(L)}$$



Transition to Linearity holds, because:

- Random initialization: each weight $w_{ij}^{(l)} \sim N(0,1)$, i,i,d.
- Each layer has the scaled summation assembly form: $\frac{1}{\sqrt{m}}\sum$ (i.e., **Obs 1**)
- Independence of sub-models hold, after an appropriate rotation. (i.e., **Obs 2**)