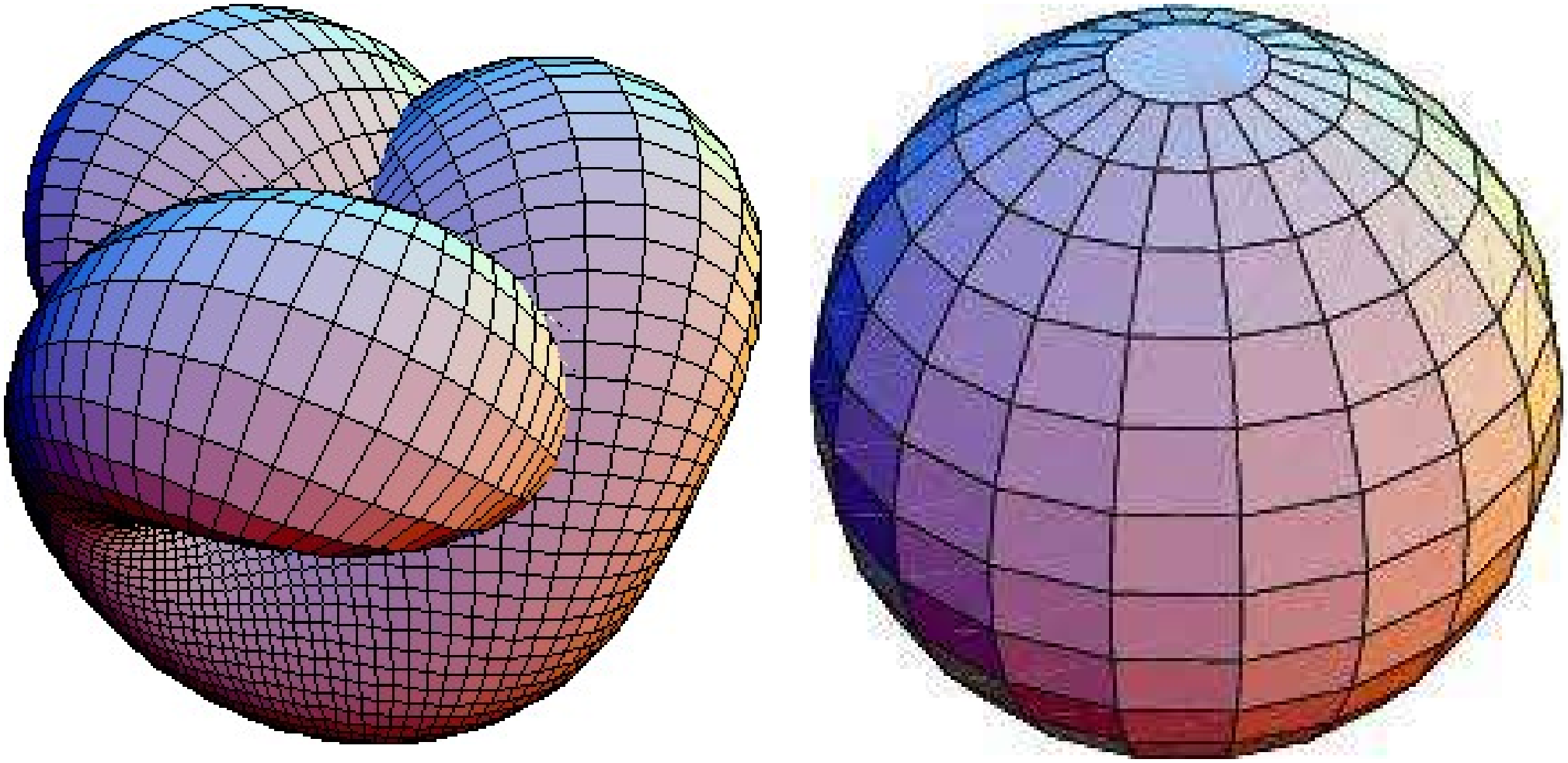
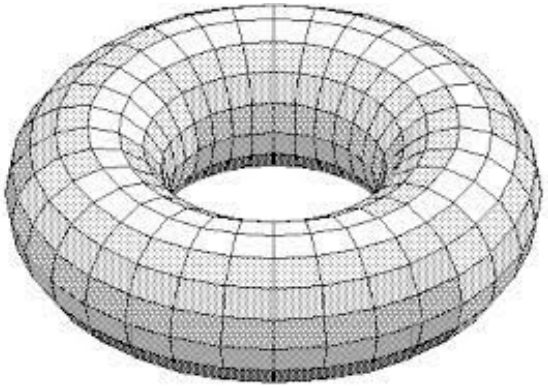
Chapter 13. Output Manifolds

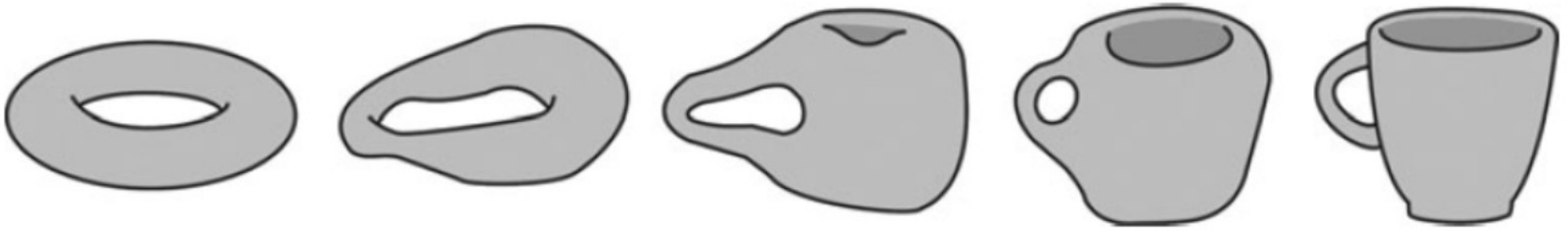



위상수학

위상수학(topology)

연결성이나 연속성 등, 작은 변환에 의존하지 않는 기하학적 성질들을 다루는 수학의 한 분야

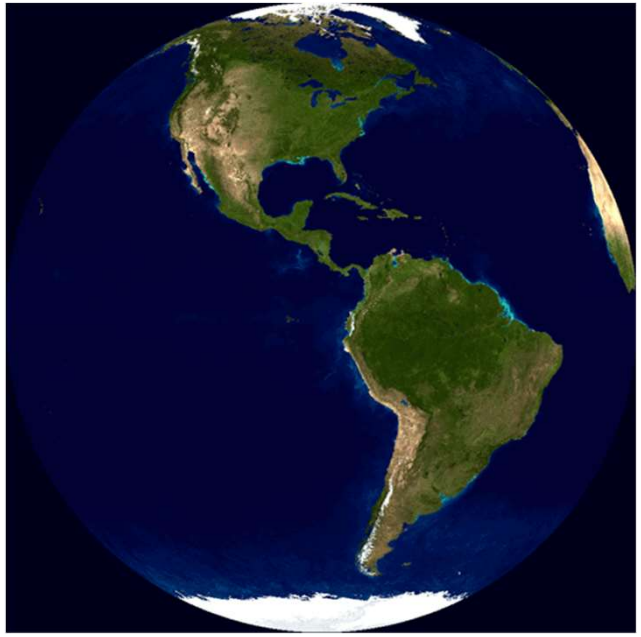




수학 좀 제법한다는 공대생 수재들이 위상수학이 뭐지? 하고 강의를 들으러 오곤 하는데, 백퍼 개거품을 물고 뒷문으로 도망간다.(그래서 우리는 뒷문을 잠궈버렸지... ㅋㅋㅋ) 해석

매니폴드??

국소적으로 유클리드 공간과 닮은 위상 공간



CHART

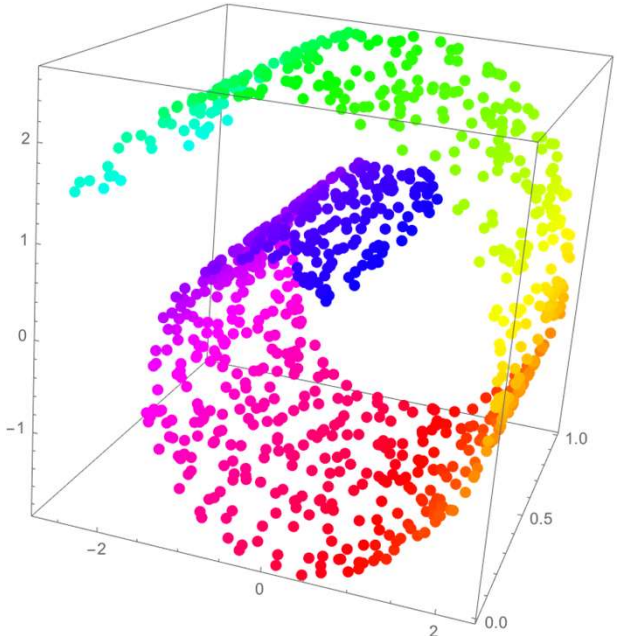
ATLAS


북극해
Arctic Ocean
유럽
Europe
아시아
Atia
북아메리카
North America
중동
Middle East
태평양
Pacific Ocean
대서양
Atlantic Ocean
아프리카
Africa
남아메리카
South America
인도양
Indian Ocean
오세아니아
Oceania
남극해
Southern Ocean

Introduction to Manifolds

A manifold is a geometrical space which resembles, at least locally, with the numerical space Rn. Each point in the manifold is described by a set of n parameters, which are considered as local coordinates. The number of parameters, n, is the dimension of the manifold. The manifold retains its own identity regardless of the parametrization.

Cloud Vision Explorer (reactive.ai)



Example 13.1.1 (Manifold of circles) The set of circles in the plane, $\mathcal{C}$, can be organized as a manifold using three parameters: the center coordinates, (a, b), and the circle radius, r. The parameters space is $\mathcal{U} = \mathbb{R}^2 \times (0, \infty)$ and the manifold parametrization is $\phi : \mathcal{U} \to \mathcal{C}$, where $\phi(a, b, r)$ is the circle centered at (a, b) and having radius r. In this case the manifold $\mathcal{C}$ is parametrized by only one map, ϕ. Each element of the manifold is a circle and the manifold has dimension 3.

Example 13.1.4 (Manifold of matrices) The set of 2×2 matrices with real entries, $\mathcal{M}_{2,2}(\mathbb{R})$, forms a 4-dimensional manifold. The parametrization is given by $\phi : \mathbb{R}^4 \to \mathcal{M}_{2,2}(\mathbb{R})$,

$$\phi(a, b, c, d) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

The set of 2×2 diagonal matrices with real entries, $\mathcal{D}_{2,2}(\mathbb{R})$, forms a 2-dimensional manifold. The parametrization is $\phi : \mathbb{R}^2 \to \mathcal{M}_{2,2}(\mathbb{R})$,

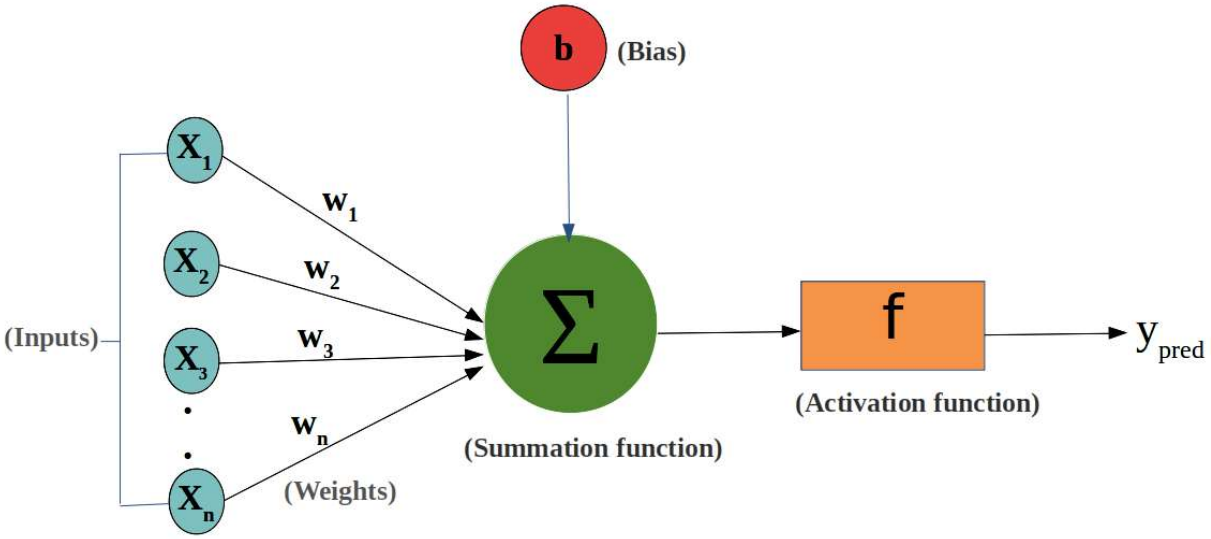
$$\phi(a, d) = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}.$$

The matrix is parametrized by only two real numbers, a and d. In fact, $\mathcal{D}_{2,2}(\mathbb{R})$ is a *submanifold* of $\mathcal{M}_{2,2}(\mathbb{R})$, as a subset which inherits the ambient manifold structure (the coordinates).

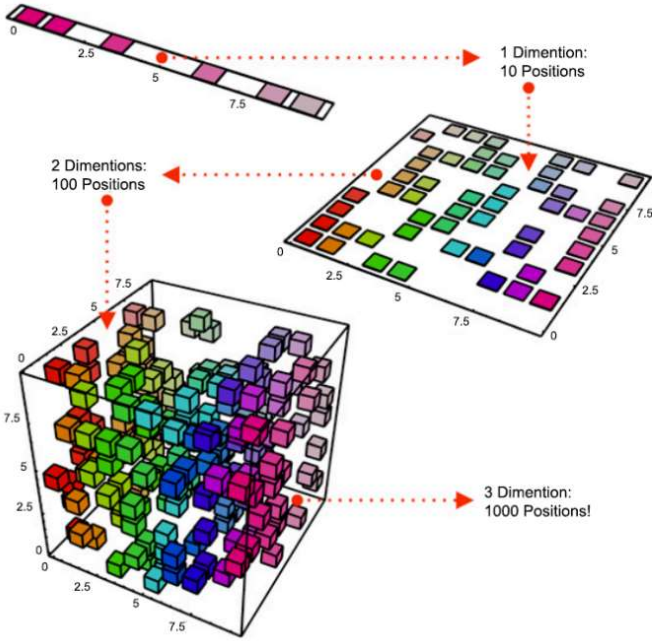
Example 13.1.6 (Manifold of sigmoid neurons) Consider a sigmoid neuron with an n-dimensional input $\mathbf{x} \in \mathbb{R}^n$ and the one-dimensional output $y = \sigma(w^T\mathbf{x}+b)$, where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the weights and the bias of the neuron. We take σ to be the logistic function. Then the set of outputs

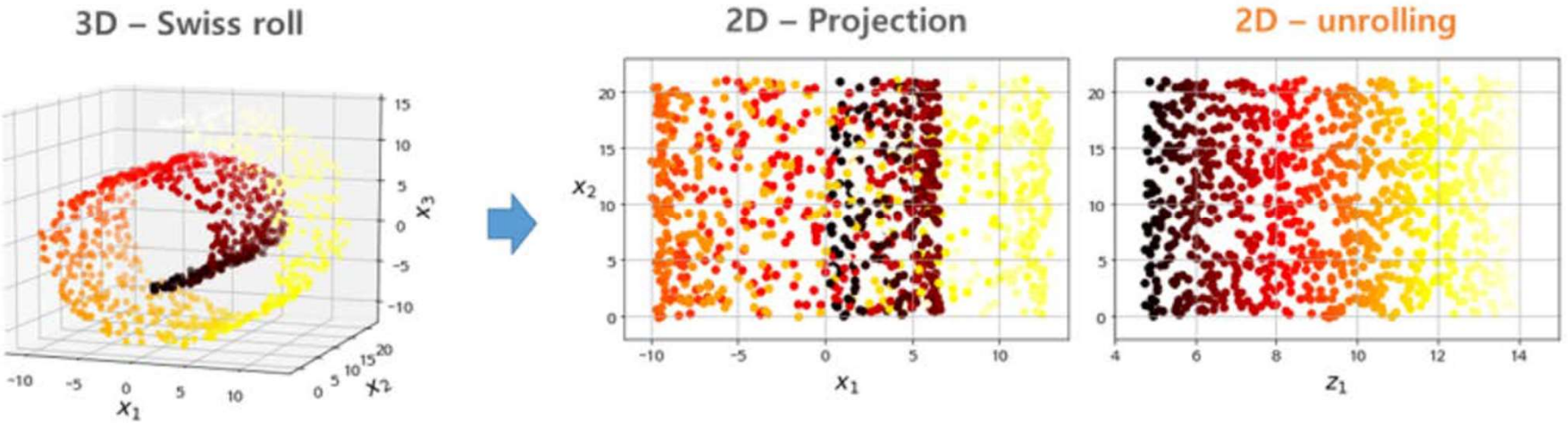
$$\mathcal{S} = \{\sigma(w^T\mathbf{x}+b); w \in \mathbb{R}^n, b \in \mathbb{R}\}$$

can be regarded as an $(n+1)$-dimensional manifold, parametrized by w and b.



차원의 저주


1 Dimention:
10 Positions
2 Dimentions:
100 Positions
3 Dimention:
1000 Positions!

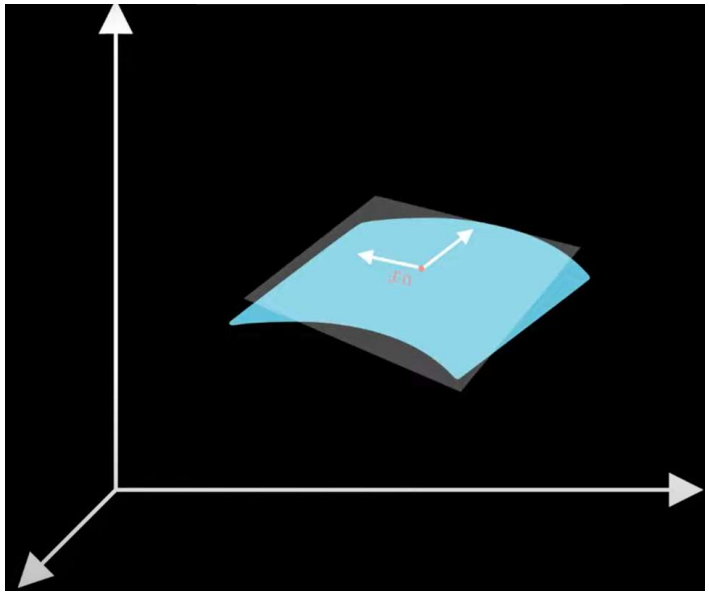

3D – Swiss roll
2D – Projection
2D – unrolling
x_1
x_2
x_3
z_1

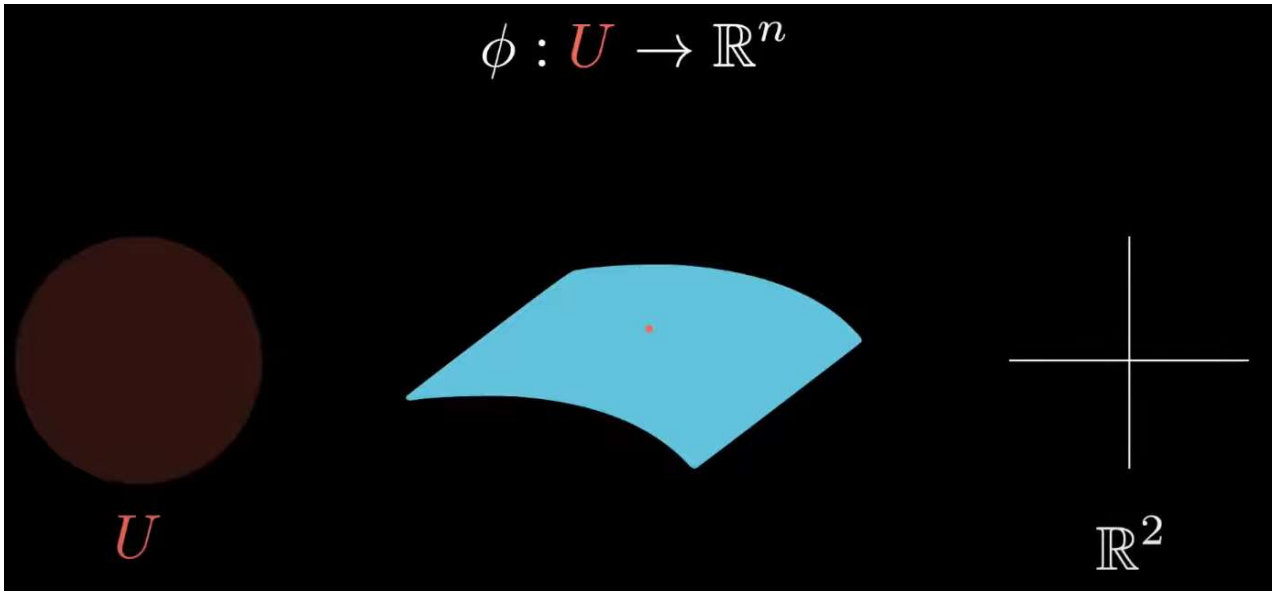
Intrinsic and Extrinsic

개미 시점: intrinsic

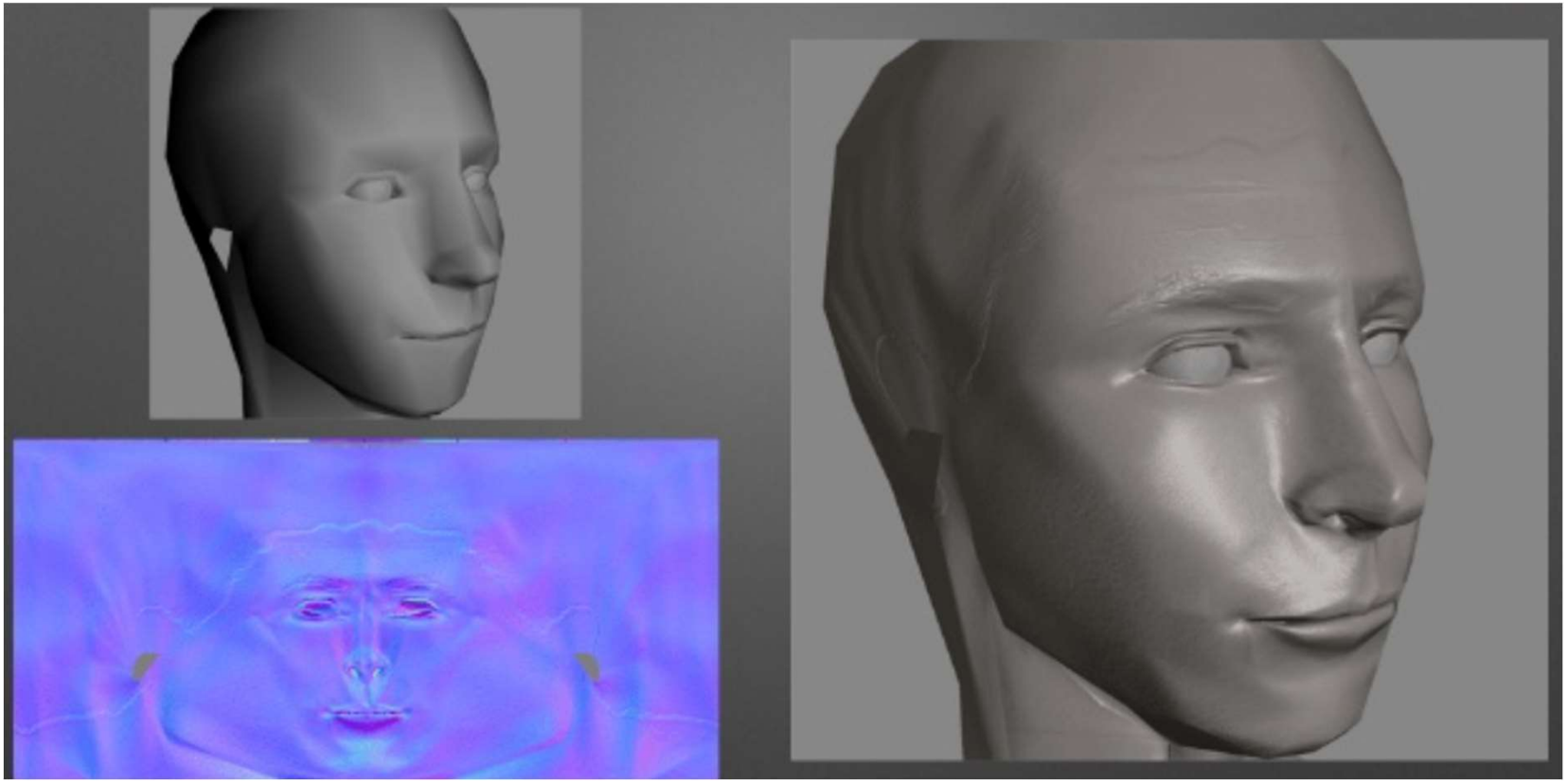
위성 관점: extrinsic

Tangent space

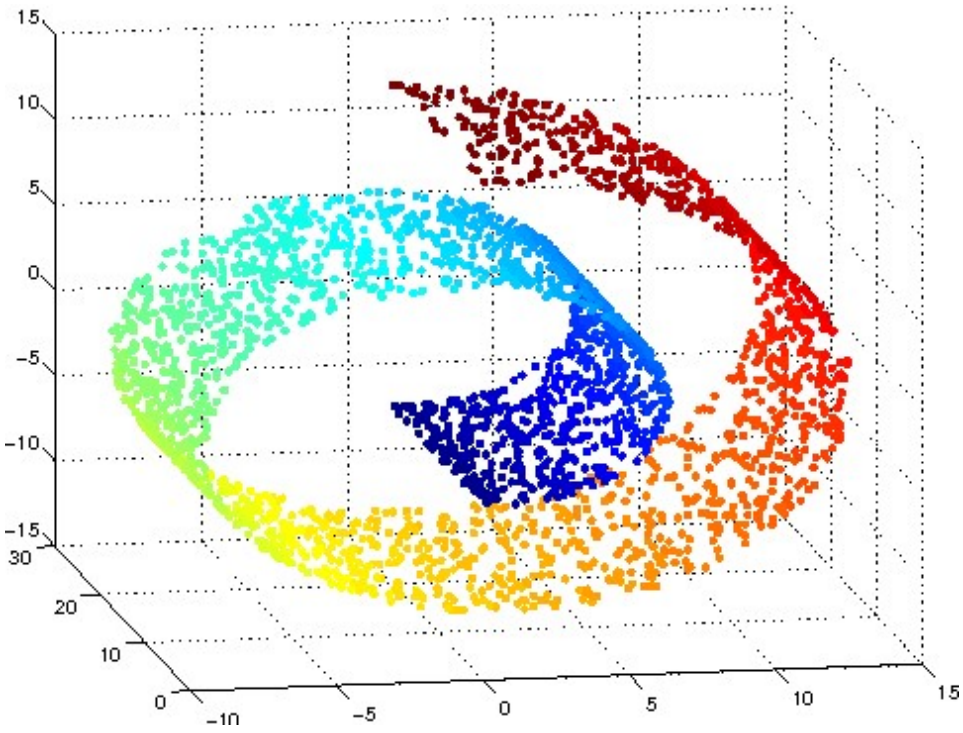




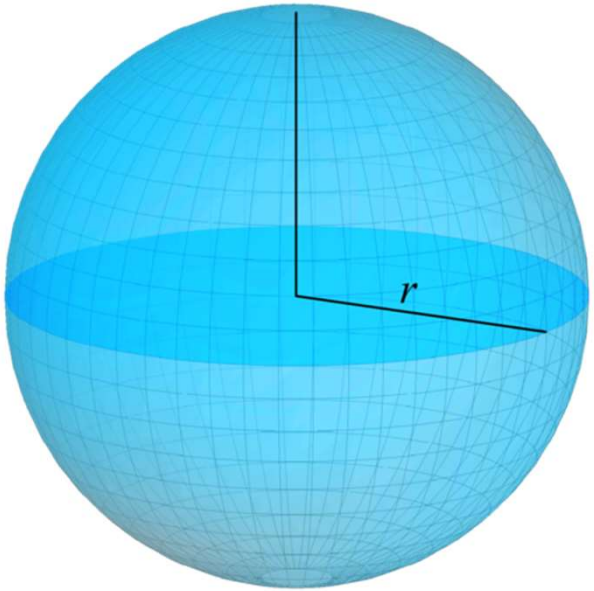
Ex) 노멀 맵

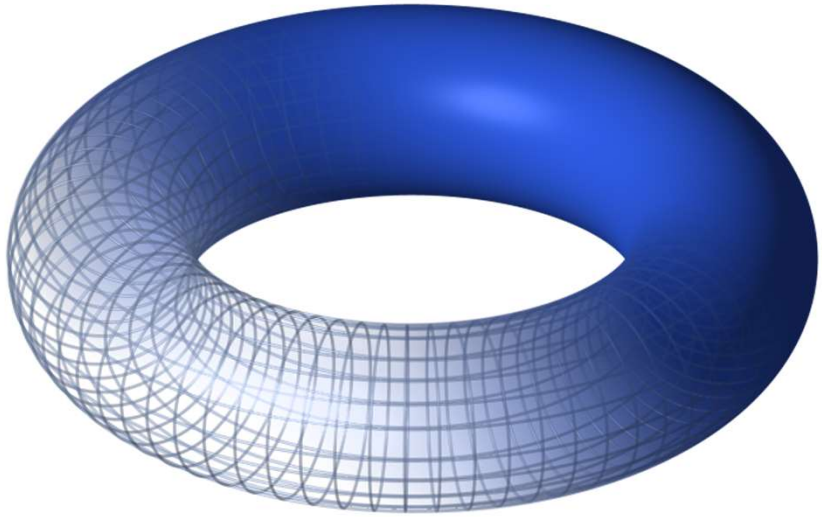


Geodesic 측지선 : 지름길



Submanifolds



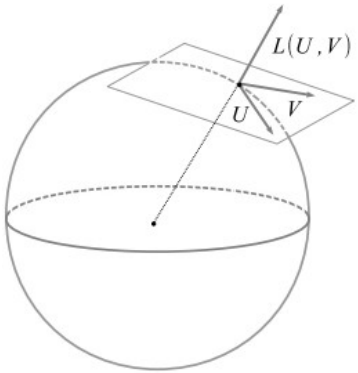


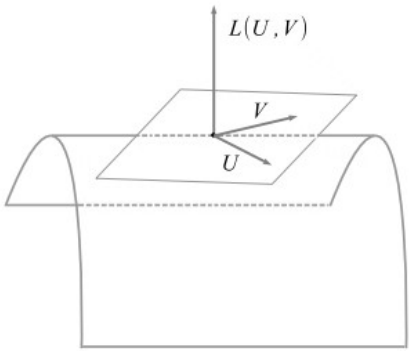
제2 기본형식

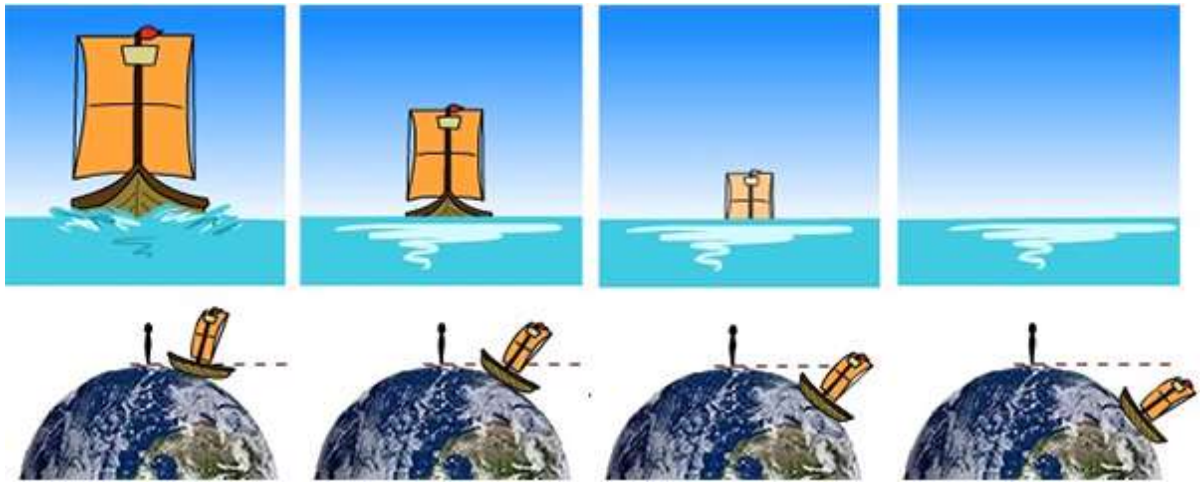
다음과 같은 **쌍선형 형식** II을 곡면 $\mathbf{x}$의 **제2 기본 형식**the second fundamental form이라 정의한다.

$$II(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{2}\sum_{j=1}^{2} L_{ij}X^iY^j = L_{ij}X^iY^j = \begin{bmatrix} X^1 & X^2 \end{bmatrix} \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Y^1 \\ Y^2 \end{bmatrix}$$

Second Fundamental Form (제2 기본형식)



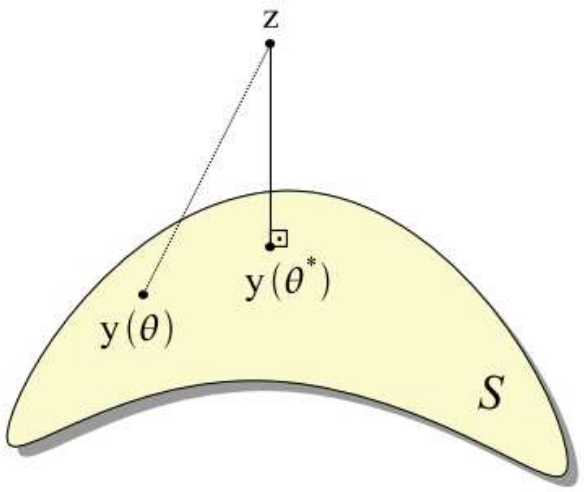




Relation to neural network

Optimal Parameter Values

$$\theta^* = \arg\min_{\theta} \operatorname{dist}(\mathbf{z}, \mathcal{S}) = \arg\min_{\theta} \|\mathbf{z} - \mathbf{y}(\theta)\|,$$



The Parameter Space

$$k = \frac{2N}{N+2}.$$

Even if this number is not always an integer, for a large number of neurons N, the optimal number of hidden layers is well approximated by $k = 2$. This explains why in the case when $N = 10$, having a network with two hidden layers, each having 5 neurons, achieves the maximum capacity.

한정된 뉴런 수 N안에서 parameter 수 최대화

$$k = \frac{2N}{d^{(0)} + d^{(L)} + N}$$

The Parameter Space

The theoretical maximum number of parameters is given by $f_N(\frac{2N}{N+2})$. In fact, this is equal to the value

$$f_N(2) = \frac{N^2}{4} + 2N.$$

Therefore, the maximum dimension of the output manifold S grows quadratically in the number of hidden neurons N.

Regularization

- Generalizing the model to avoid overfitting

1. Parsimony criterion

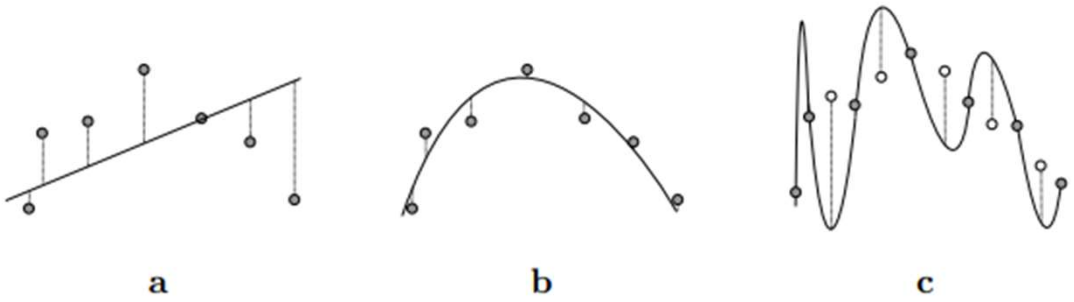


Figure 13.13: *Polynomial regression through 7 points:* (a) *Using a line leads to an underfit;* (b) *using a quadratic polynomial leads to a good fit;* (c) *the use of a 7th degree polynomial overfits the data.*

$$k = \frac{2N}{d^{(0)} + d^{(L)} + N}$$

Hidden layer 수 k=2일 때 parameter 수 최대
→k=1, or 3,4,5,..

2. Norm regularization

$$L(w) = C(w, b) + \lambda \|w\|^2$$

3. Choosing the flattest manifold

- Which structure of S is better from the regularization point of view?

$$C(w, b; \mu) = \frac{1}{2}\|\mathbf{y}(w, b) - \mathbf{z}\|^2 + \mu\|L\|$$

3. Choosing the flattest manifold

- Second fundamental form L ≈ 표면 굽은 정도

  표면 normal vector 얼마나 빨리 변하는지

3. Choosing the flattest manifold

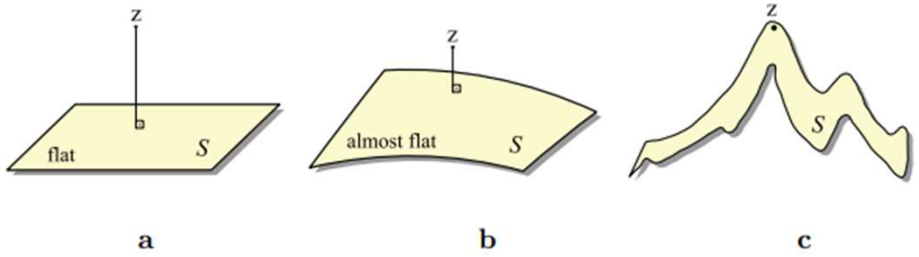


Figure 13.14: *Regularization with a manifold $\mathcal{S}$ of the same dimension but different degrees of flatness:* (a) *Using a plane leads to a large distance from the target* $\mathbf{z}$*, which underfits data;* (b) *using a trade-off between curvature and distance to* $\mathbf{z}$ *leads to a good fit;* (c) *using a largely curved manifold we can always force the target point* $\mathbf{z}$ *to belong to the manifold, case which corresponds to an overfit.*

Example 13.6.1 (polynomial regression)

$$\psi(x;\theta) = x^r + \theta_1 x^{r-1} + \theta_2 x^{r-2} + \cdots + \theta_{r-1}x + \theta_r \qquad L = 0$$

Example 13.6.2 (single sigmoid neuron)

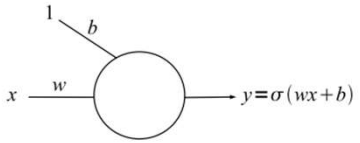


Figure 13.15: *The manifold $\mathcal{S}$ associated with a sigmoid neuron is 2-dimensional.*

Can do something about L

4. Model averaging

Figure 13.16: *Two neural nets with the same input,* $\mathbf{x}$*, and outputs* $\mathbf{y}(w, b)$*,* $\widetilde{\mathbf{y}}(\widetilde{w}, \widetilde{b})$*, learning the same target* $\mathbf{z}$*.*

4. Model averaging

- 변형: convex combination 사용

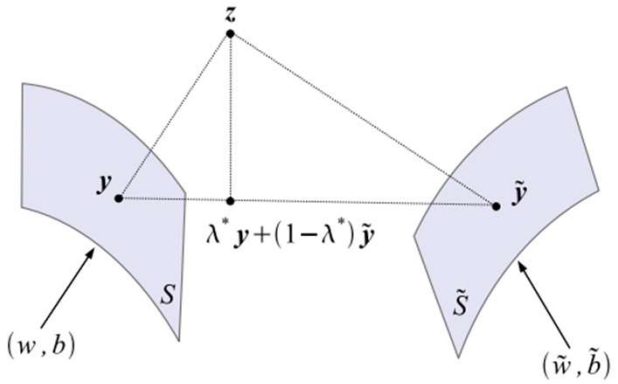


Figure 13.17: *The orthogonal projection of* $\mathbf{z}$ *on the line segment* $\mathbf{y}\tilde{\mathbf{y}}$ *is a better approximator than both* $\mathbf{y}$ *and* $\tilde{\mathbf{y}}$. *This is given by* $\lambda^*\mathbf{y} + (1-\lambda^*)\tilde{\mathbf{y}}$, where λ^* *is obtained as* $\lambda^* = \arg\min \|\mathbf{z} - \lambda\mathbf{y} - (1-\lambda)\tilde{\mathbf{y}}\|$.

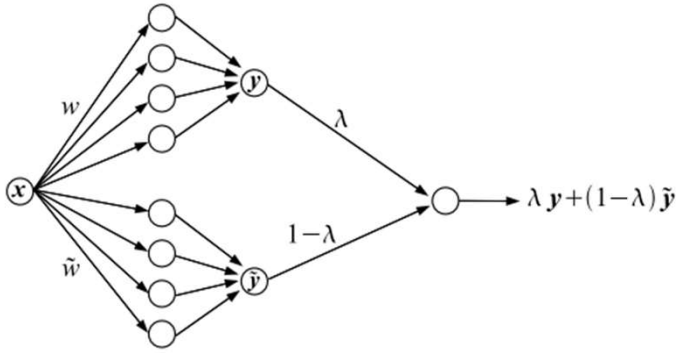


Figure 13.18: *The model combination of two nets is a net that produces a better learning than both of its parts.*

5. Dropout

- Large family of neural net 일 때 효율적인 regularization
- Coadaptation 방지
- L_2 norm regularization과 유사

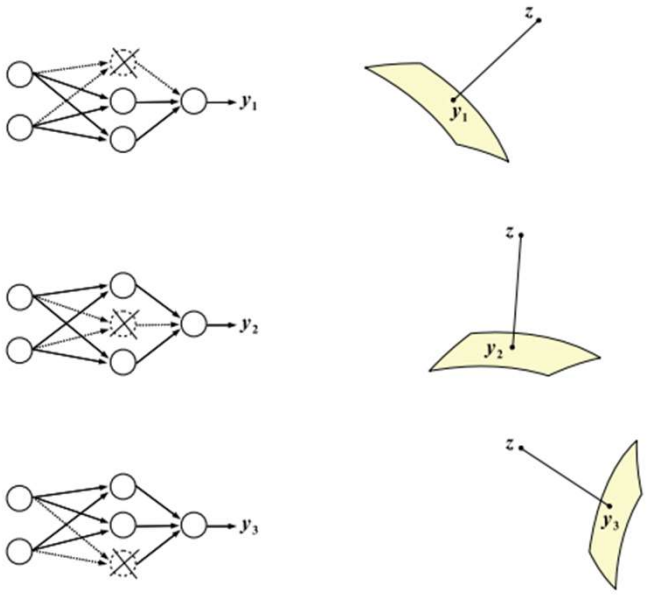


Figure 13.19: *When one neuron is dropped at a time, the network output produces projections of the target* $\mathbf{z}$ *onto the associated output manifolds. The average of projections,* $\frac{1}{3}(\mathbf{y}_1+\mathbf{y}_2+\mathbf{y}_3)$*, is supposed to be a better approximation of* $\mathbf{z}$ *than any of the* $\mathbf{y}_j$.

5. Dropout

- 일반화

in the case of dropout becomes

$$x_j^{(\ell)} = \phi\Big(\sum_{i=1}^{d^{(\ell-1)}} w_{ij}^{(\ell)} \tilde{x}_i^{(\ell-1)} - b_j^{(\ell)} \Big), \qquad 1 \le j \le d^{(\ell)},$$

where $\tilde{x}_i^{(\ell)} = R_i^{(\ell)} x_i^{(\ell)}$, with $R_i^{(\ell)} \sim \text{Bernoulli}(p)$

Bernoulli(p)말고도 Uniform, Gaussian 분포 따르는 R 곱해서 인위적 noise 넣어 regularization 가능하다

Remark 13.6.4 Empirical evidence has shown that the optimal retention rate for hidden layers is usually $p = 0.5$, while for the input layer is about $p = 0.8$.

5. Dropout

- L_2 norm regularization과 유사
- 교재 p.460

Linear regression with dropout This section deals with the application of dropout in the case of the classical problem of linear regression. Consider the input vector $X \in \mathbb{R}^n$ and the target $\mathbf{z} \in \mathbb{R}$. We need to learn the weights vector $\mathbf{w} \in \mathbb{R}^n$ such that $\|\mathbf{z} - X\mathbf{w}\|^2$ is minimized. Applying dropout, the new objective function becomes

$$f(\mathbf{w}) = \mathbb{E}[\|\mathbf{z} - R \odot X\mathbf{w}\|^2],$$

where $R^T = (R_1, \ldots, R_n)$ is a vector of independent Bernoulli random variables, $R_i \sim \text{Bernoulli}(p)$. Using that $\|a - b\|^2 = \|a\|^2 - 2a^T b + \|b\|^2$ and the linearity of the expectation operator, the objective function becomes

$$\begin{aligned}
f(\mathbf{w}) &= \|\mathbf{z}\|^2 - 2\mathbf{z}^T\mathbb{E}[R] \odot X\mathbf{w} + \mathbb{E}[\|R \odot X\mathbf{w}\|^2] \\
&= \|\mathbf{z}\|^2 - 2p\mathbf{z}^T X\mathbf{w} + p^2\|X\mathbf{w}\|^2 + \mathbb{E}[\|R \odot X\mathbf{w}\|^2] - p^2\|X\mathbf{w}\|^2 \\
&= \|\mathbf{z} - pX\mathbf{w}\|^2 + \mathbb{E}[\|R \odot X\mathbf{w}\|^2] - p^2\|X\mathbf{w}\|^2 \\
&= \|\mathbf{z} - pX\mathbf{w}\|^2 + \mathbb{E}[\mathbf{w}^T(R \odot X)^T(R \odot X)\mathbf{w}] - p^2\|X\mathbf{w}\|^2 \\
&= \|\mathbf{z} - pX\mathbf{w}\|^2 + \mathbb{E}[R^2]\mathbf{w}^T X^T X\mathbf{w} - p^2\|X\mathbf{w}\|^2 \\
&= \|\mathbf{z} - pX\mathbf{w}\|^2 + p\mathbf{w}^T X^T X\mathbf{w} - p^2\|X\mathbf{w}\|^2 \\
&= \|\mathbf{z} - pX\mathbf{w}\|^2 + p(1-p)\|X\mathbf{w}\|^2,
\end{aligned}$$

where we have added and subtracted the term $p^2\|X\mathbf{w}\|^2$ to form a square of the norm and used that the second moment of a Bernoulli variable is p.

Absorbing the factor p into the weight $\mathbf{w}$, the objective function becomes

$$f(w) = \|\mathbf{z} - Xw\|^2 + \lambda\|Xw\|^2,$$

which is an L^2-regularization problem with the Lagrange multiplier $\lambda = \frac{1-p}{p}$ and $w = p\mathbf{w}$. When p tends to 1, all neurons are retained and λ gets small. The constant λ represents the ratio between the non-retained and the retained neurons during the dropout process. Hence, a linear regression with dropout is equivalent to a L^2-regularization problem.

14. Neuromanifolds

학습목표

- Ch 13, Ch14 관점 차이 이해하기
- Cross entropy loss= KL Divergence ≈ negative log likelihood 이해하기
- The natural gradient learning과 simulated annealing method 에 대해 들어보기

Intro

- In Chapter13

In this chapter we shall associate a manifold with each neural network by considering the weights and biasses of a neural network as the coordinate system on the manifold. This manifold can be endowed with a Riemannian metric,

- In Chapter 14

In this chapter we shall approach the study of neural networks from the Information Geometry perspective. This applies both techniques of Differential Geometry and Probability Theory to neural networks.

The difference from the theory introduced in Chapter 13 is that here the network's input and target are probability densities of random variables and the neural network output contains some noisy perturbation. This way, the family of joint probability densities of the input and output, $p(x, y; \theta)$, becomes a statistical manifold, which is parametrized by θ; thus, the weights and biasses play the role of a coordinate system for the associated statistical manifold. The intrinsic distance between two neural networks is measured in this space using the Fisher information metric. Roughly speaking, the Fisher metric represents the amount of information about network's own weights and biasses that is contained in the training distribution. The associated statistical manifold endowed with the Fisher metric becomes a Riemannian manifold, called a neuromanifold.

1. Statistical Manifolds

- Neural network f
- Input : random variable X
- Output variable $Y = f_\theta(X)$
- Network parameter $\theta = (w, b)$
- Input dist. $p_X(x)$
- Output dist. $p_Y(y; \theta)$
- Joint input-output dist. $p(x, y; \theta)$

- Target: random variable Z
- Joint dist. $p(x, z)$: training distribution
- $Z = Y + \epsilon(\theta)$
- When mean square cost function used: $C(\theta) = \frac{1}{2}\mathbb{E}[(Z - Y)^2] = \frac{1}{2}\mathbb{E}[\epsilon(\theta)^2].$

1. Statistical Manifolds

- Noisy neurons : regularization
  $Y = f_\theta(X) + n$

Ex) $n \sim N(0,1)$

$$p(y|x;\theta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y - f_\theta(x))^2}$$

Ex) $n \sim Unif(-1,1)$

$$p(y|x;\theta) = \begin{cases} \frac{1}{2}, & \text{if } f_\theta(x) - 1 \le y \le f_\theta(x) + 1 \\ 0, & \text{otherwise.} \end{cases}$$

$$p(x, y; \theta) = p(x)p(y|x;\theta)$$

- Goal: tune the parameter θ s.t. $p(x, y; \theta)$ matches as much as possible the true dist. $p(x, z)$

1. Statistical Manifolds

- Family of density function $\{\theta \to p(x,y;\theta);\ \theta \in \Theta\}$ Θ: N-dim parameter space

:submanifold of infinite-dim space of pdfs

- Assume regularity condition

$$\frac{\partial}{\partial\theta_1}p(x,y;\theta),\ldots,\frac{\partial}{\partial\theta_N}p(x,y;\theta)$$ are linearly independent

:submanifold is smooth and admits a tangent space at each point $p(x,y;\theta)$

→The manifold S= $\{p(x,y;\theta);\theta\}$ is called statistical manifold

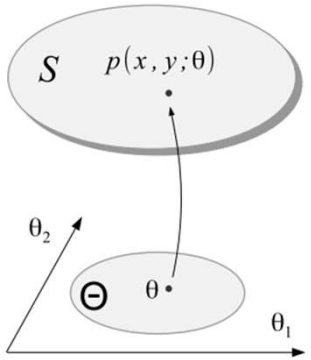


Figure 14.1: *A statistical manifold* $\mathcal{S} = \{p(x,y;\theta);\theta \in \Theta\}$

1. Statistical Manifolds

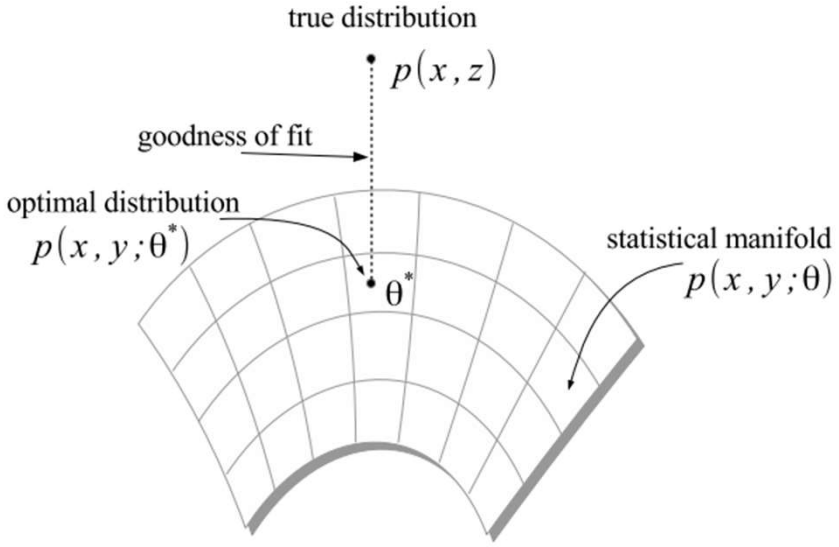


Figure 14.2: *Geometric image of the goodness of fit (loss function).*

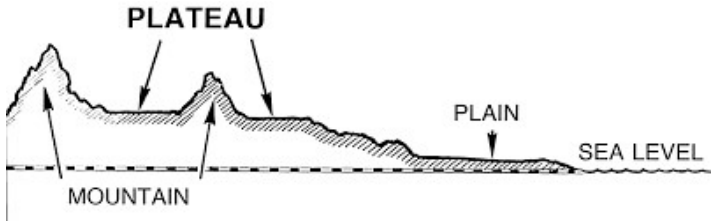
Any cost function can be considered, but Kullback-Leibler divergence is preferred due to its relation to the maximum likelihood estimation.

argmin θ cross entropy loss
=argmin θ KL divergence
=(given empirical, mutually independent data(x′,z′)
θ: MLE of $p(x, z; \theta)$

(교재 3.6장, p.468)

1. Statistical Manifolds

- 점과 점 / 분포 맞춘다
- MSE / CE less plateaus, time efficient



2. Fisher metric

- Fisher information

$$I(\theta) = \mathbb{E}\Big[\Big(\frac{\partial \ell(\theta)}{\partial \theta}\Big)^2\Big] \qquad g_{ij}(\theta) = \mathbb{E}\Big[\frac{\partial \ell(\theta)}{\partial \theta_i}\frac{\partial \ell(\theta)}{\partial \theta_j}\Big]$$

Under regularity condition

$\frac{\partial}{\partial \theta_1} p(x, y; \theta), \ldots, \frac{\partial}{\partial \theta_N} p(x, y; \theta)$ are linearly independent

$g_{ij}(\theta)$ is symmetric, positive definite, nondegenerate matrix

→ Fisher information matrix : Riemannian metric on the statistical manifold S={$\theta \to p(x; \theta)$}
Now called Fisher metric.
(Riemannian metric g_{ij} : weights used in measuring the distance between the points on the manifold)

$$d(P, P') = \Big(\sum_{i,j=1}^{n} g_{ij}(\Delta x)_i (\Delta x)_j\Big)^{1/2} \qquad \mathbb{R}^n : \; g_{ij} = \delta_{ij} \; : \; d(P, P') = \Big(\sum_{i=1}^{n} (\Delta x)_i^2\Big)^{1/2}$$

2. Fisher metric

- What makes the Fisher metric distinguishable among all Riemannian metrics that can be defined on a statistical manifold?
- Unique satisfying both conditions

1. g_{ij} is invariant under reparametrizations of the sample space. This means the statistical manifolds $\mathcal{S} = \{p(x;\theta);\theta \in \Theta\}$ and $\tilde{\mathcal{S}} = \{p(h(x);\theta);\theta \in \Theta\}$, with h invertible and differentiable function, have equal metrics, $g_{ij}(\theta) = \tilde{g}_{ij}(\theta)$. This invariance property can be found in Theorem 1.6.4 of [22].

2. g_{ij} is covariant under reparametrizations. This means that if we consider a different parametrization $\xi_j = \xi(\theta_1, \ldots, \theta_N)$ depending on θ, the Fisher matrices in both parametrizations are related by the relation:

$$g_{ij}(\theta) = \sum_{k.r} g_{kr}(\xi)\Big|_{\xi=\xi(\theta)} \frac{\partial \xi^k}{\partial \theta^i} \frac{\partial \xi^r}{\partial \theta^j}.$$

3. Neuromanifold

Definition 14.3.1 *The neuromanifold associated with the aforementioned neural network is the Riemannian manifold* $(\mathcal{S}, g)$*, where* $\mathcal{S}=\{p(x,y;\theta);\theta\in\Theta\}$ *is the statistical manifold of the joint input-output densities of the neural network,* θ *are the network weights and biasses, and* g *is the Fisher metric.*

4. The natural gradient learning

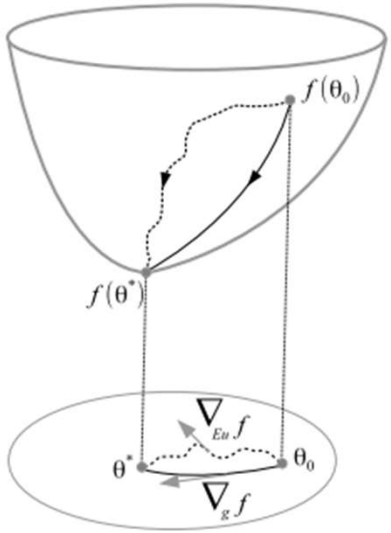


Figure 14.4: *The natural gradient descent arrives faster to the minimum than the Euclidean gradient descent method does.*

$$\nabla_g f = g(\theta)^{-1} \nabla_{Eu} f$$

$$\theta_{n+1} = \theta_n - \eta_n \nabla_g C(\theta_n)$$

$$-\nabla_g C(\theta) = -g^{-1}(\theta) \nabla_{Eu} C(\theta)$$

$$\theta_{n+1} = \theta_n - \eta_n g(\theta)^{-1} \nabla_{Eu} C(\theta_n)$$

$$g_{ij}(\theta) = \mathbb{E}\left[\frac{\partial \ell(\theta)}{\partial \theta_i} \frac{\partial \ell(\theta)}{\partial \theta_j}\right]$$

4. The natural gradient learning

- No trapped in plateau
- Asymptotically Fisher efficient in online learning

-batch learning

-Fisher efficient

-adaptive implementation

• $\hat{\theta}$ is *Fisher-efficient* if it is unbiased and reaches the lower bound in the Cramér-Rao inequality

$$Cov(\hat{\theta}) \geq g^{-1}(\theta), \quad \forall \theta \in \Theta,$$

i.e., it is a minimum variance unbiased estimator.[2]

$$\begin{aligned} \hat{g}_{n+1}^{-1} &= (1+\epsilon_n)\hat{g}_n^{-1} - \epsilon_n \hat{g}_n^{-1} \nabla_{Eu} f_n (\nabla_{Eu} f_n)^T \hat{g}_n^{-1} \\ \theta_{n+1} &= \theta_n - \eta_n \hat{g}_n^{-1} \nabla_{Eu} \ell(x_n, z_n; \theta_n), \end{aligned}$$

5. Simulated annealing method

- Natural gradient learning
- Global minimum

Increasing resolution method

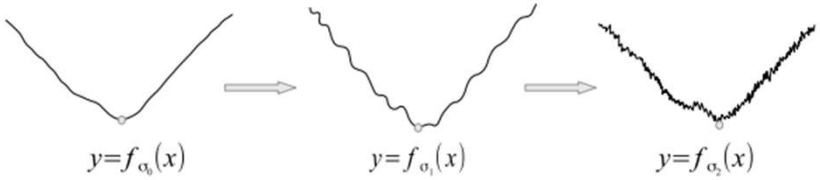


Figure 4.17: *The increase in the signal resolution. The minimum on each resolution profile is represented by a little circle.*

$$n_T \sim \mathcal{N}(0, T^2)$$

$$T_1 > T_2 > \cdots > T_N > 0.$$

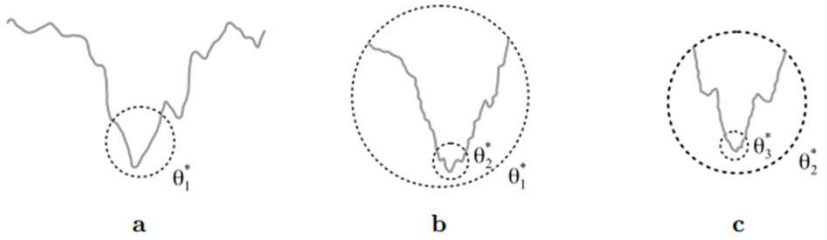
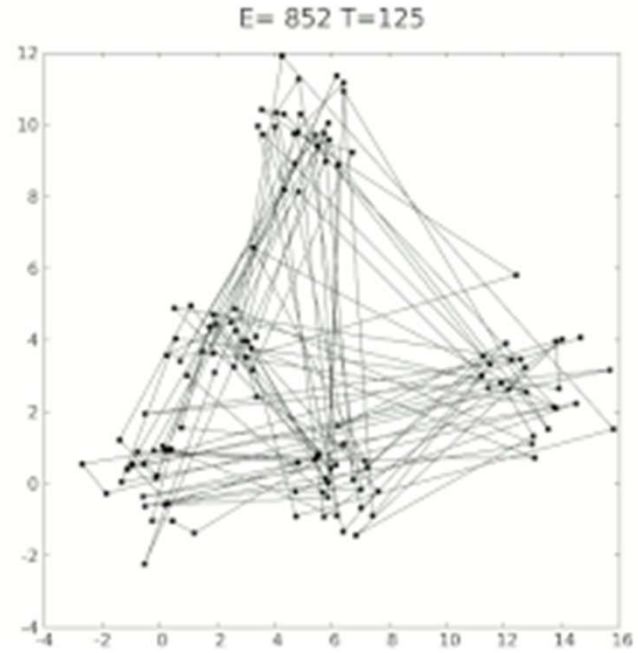
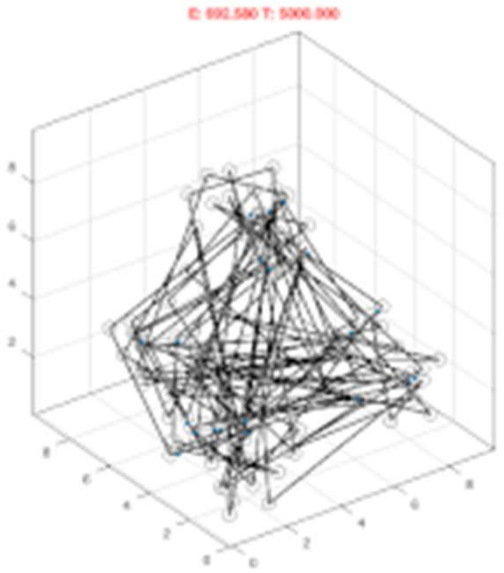


Figure 14.6: *Annealing method:* **a**. *For a large temperature, T_1, the optimal parameter θ_1^* is located in a neighborhood of the global minimum.* **b**. *Decreasing the temperature to T_2, we obtain a more accurate approximation of the global minimum given by the new optimum value, θ_2^*.* **c**. *Continuing to decrease temperature we obtain more and more accurate approximations of the global minimum.*

5. Simulated annealing method

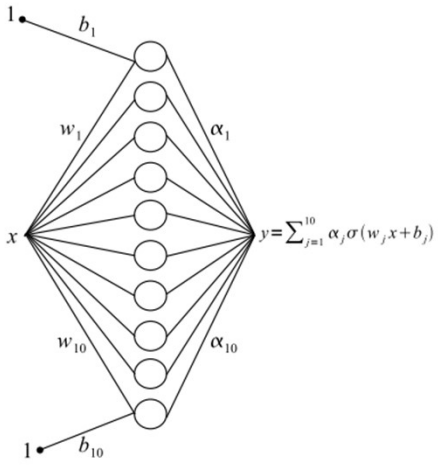
- Traveling salesman problem
- en.wikipedia.org/wiki/Simulated_annealing

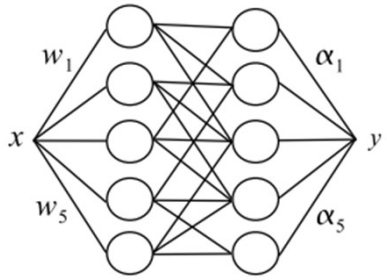


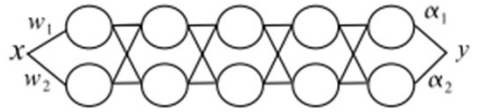


HW

- Input neuron, output neuron 개수가 하나 씩이라고 가정하고 hidden layer에 N=10개의 neuron을 쓴다고 할 때 hidden layer의 개수가 1개, 2개, 5개일 때 neural network의 총 parameter 수(weight par.+bias par.)를 구하시오.







HW

Exercise 13.8.1 A feedforward neural network of type 784-200-100-50-10 is used to classify the MNIST data. Find the dimension of the associated output manifold. (784 is the input size and 10 represents the number of digit classes).

Let MNIST training data: 55000개
Let overfit: total parameter수>training data 수

Exercise 13.8.3 A one-hidden layer feedforward neural net, 784-N-10, is used to classify the MNIST data. Find the range of the number of hidden neurons, N, for which the network overfits the training data.

Exercise 13.8.4 A two-hidden layer feedforward neural net, 784-h-h-10, is used to classify the MNIST data. Find the range of the number h, for which the network overfits the training data.

HW

Exercise 13.8.10 Consider the model combination of two sigmoid neurons. Write the output of the combination and specify the dimension of the associated output manifold.

Exercise 13.8.11 List a few effects of dropping neurons from a network on the associated output manifold.