Key Features of the Error Surface

■ Local Minima

○ Nonlinear networks usually have multiple local minima of differing depths. The goal of training is to locate one of these minima.

○ Random(Multiple) Restarts: One of the simplest ways to deal with local minima is to train many different networks with different initial weights.

■ Flat Regions (Saddle Points)

■ High-Dimensional

Implications for Training

■ Possibly Questionable Solution Quality.

○ The optimization process may or may not find a good solution and solutions can only be compared relatively, due to deceptive local minima.

■ Possibly Long Training Time.

○ The optimization process may take a long time to find a satisfactory solution, due to the iterative nature of the search.

■ Possible Failure.

○ The optimization process may fail to progress (get stuck) or fail to locate a viable solution, due to the presence of flat regions.

Components of the Learning Algorithm

Network Topology
Loss Function.
Weight Initialization.
Batch Size.
Learning Rate.
Epochs.
Data Preparation.

Gradient Decent

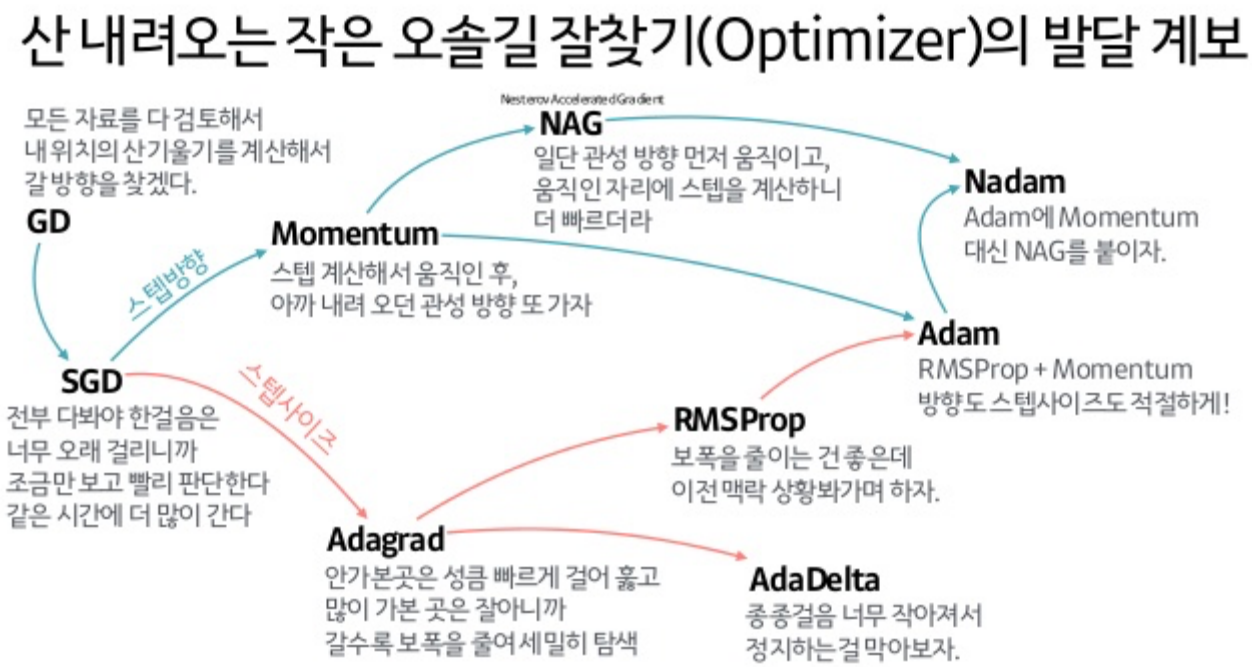
■ 참고

○ http://ruder.io/optimizing-gradient-descent/index.html

○ https://remykarem.github.io/blog/gradient-descent-optimisers.html

■ 기본 개념

$$w_{\text{new}} = w - \alpha \frac{\partial L}{\partial w}$$



[그림 2] 출처: 하용호, 자습해도 모르겠던 딥러닝, 머리속에 인스톨 시켜드립니다

Optimiser	Year	Learning Rate	Gradient
Momentum	1964		✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	
Nesterov	2013		✓
Adam	2014	✓	✓
AdaMax	2015	✓	✓
Nadam	2015	✓	✓
AMSGrad	2018	✓	✓

■ 확률적 경사 하강법(SGD)

경사 하강법의 불필요하게 많은 계산량은 속도를 느리게 할 뿐 아니라, 최적 해를 찾기 전에 최적화 과정이 멈출 수도 있습니다. 확률적 경사 하강법(Stochastic Gradient Descent, SGD)은 이러한 속도의 단점을 보완한 방법입니다. 전체 데이터를 사용하는 것이 아니라, 랜덤하게 추출한 일부 데이터를 사용합니다. 일부 데이터를 사용하므로 더 빨리 그리고 자주 업데이트를 하는 것이 가능해졌습니다.

그림 9-5는 경사 하강법과 확률적 경사 하강법의 차이를 보여 줍니다. 랜덤한 일부 데이터를 사용하는 만큼 확률적 경사 하강법은 중간 결과의 진폭이 크고 불안정해 보일 수도 있습니다. 하지만 속도가 확연히 빠르면서도 최적 해에 근사한 값을 찾아낸다는 장점 덕분에 경사 하강법의 대안으로 사용되고 있습니다.

■ 모멘텀

모멘텀(momentum)이란 단어는 관성, 탄력, 가속도라는 뜻입니다. 모멘텀 SGD란 말 그대로 경사 하강법에 탄력을 더해 주는 것''입니다. 다시 말해서, 경사 하강법과 마찬가지로 매번 기울기를 구하지만, 이를 통해 오차를 수정하기 전 바로 앞 수정 값과 방향(+, -)을 참고하여 같은 방향으로 일정한 비율만 수정되게 하는 방법입니다. 따라서 수정 방향이 양수(+) 방향으로 한 번, 음수(-) 방향으로 한 번 지그재그로 일어나는 현상이 줄어들고, 이전 이동 값을 고려하여 일정 비율만큼만 다음 값을 결정하므로 관성의 효과를 낼 수 있습니다.

구분	개요	효과
Stochastic Gradient Descent (SGD)	랜덤하게 추출한 일부 데이터를 사용해 더 빨리, 자주 업데이트를 하게 하는 것	**속도 개선**
Momentum	관성의 방향을 고려해 진동과 폭을 줄이는 효과	**정확도 개선**
Nesterov Accelerated Gradient (NAG)	모멘텀이 이동시킬 방향으로 미리 이동해서 그레이디언트를 계산. 불필요한 이동을 줄이는 효과	**정확도 개선**
Adagrad	변수의 업데이트가 잦으면 학습률을 적게 하여 이동 보폭을 조절하는 방법	**보폭 크기 개선**
RMSProp	아다그라드의 보폭 민감도를 보완한 방법	**보폭 크기 개선**
Adam	모멘텀과 알엠에스프롭 방법을 합친 방법	**정확도와 보폭 크기 개선**

keras.optimizers.SGD(lr=0.1)

keras.optimizers.SGD(lr=0.1, momentum=0.9)

keras.optimizers.SGD(lr=0.1, momentum=0.9, nesterov=True)

keras.optimizers.Adagrad(lr=0.01, epsilon=1e-6)

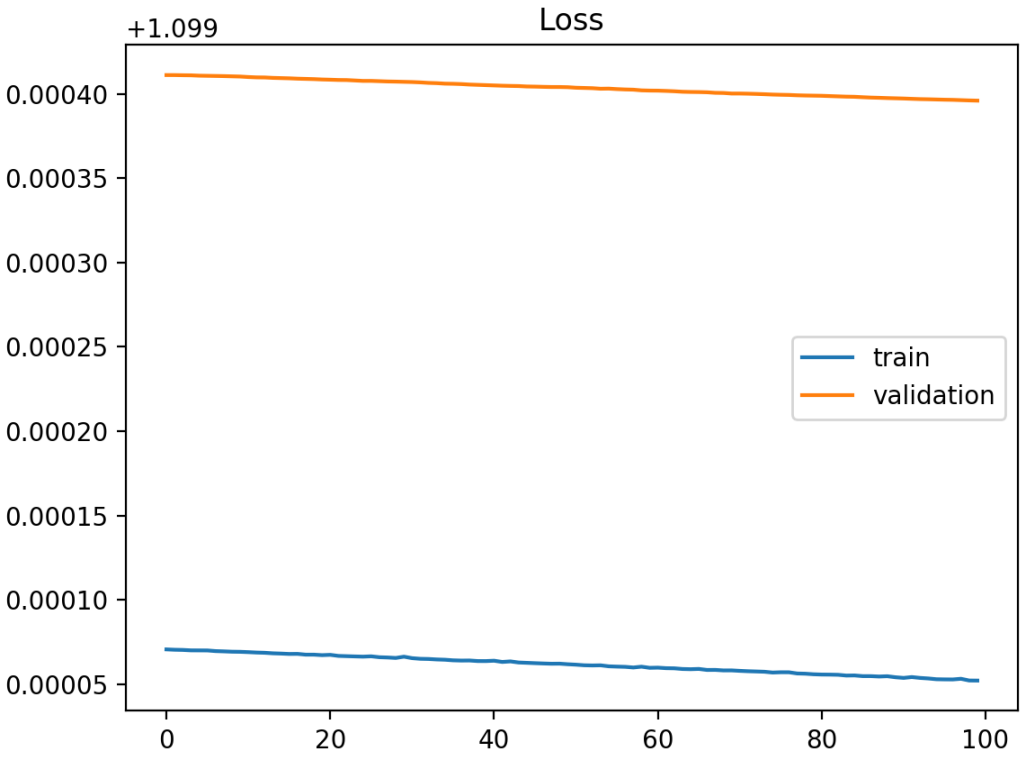
※ 참고: 여기서 epsilon, rho, decay 같은 파라미터는 바꾸지 않고 그대로 사용하기를 권장

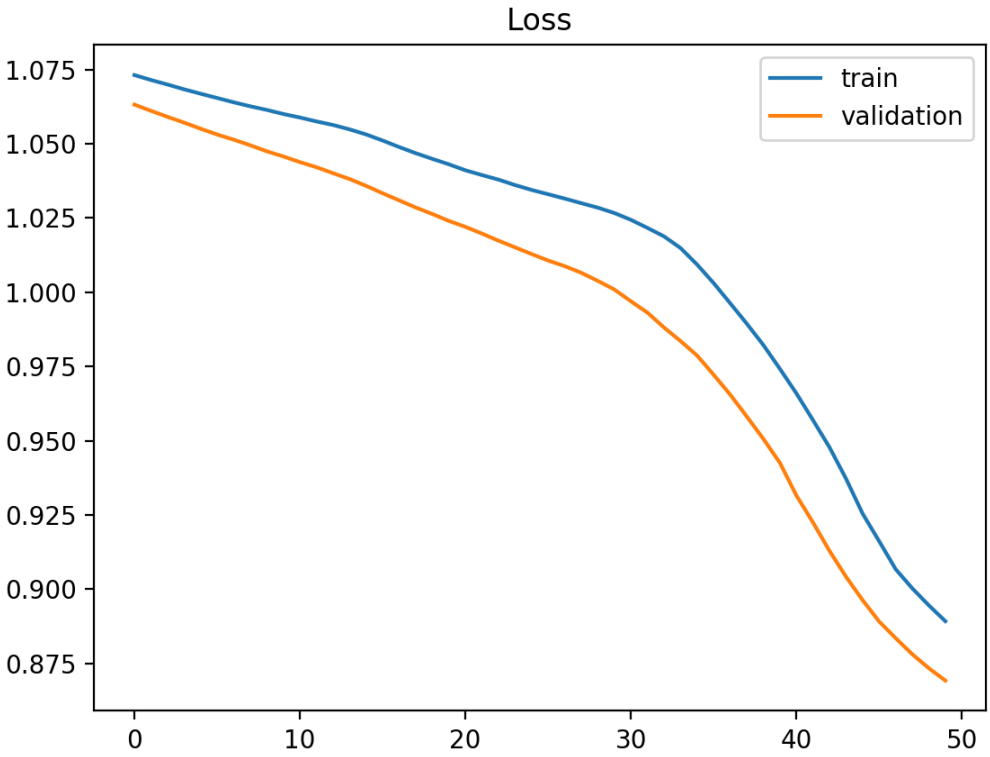
keras.optimizers.RMSprop(lr=0.001, rho=0.9, epsilon=1e-08, decay=0.0)

keras.optimizers.Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e - 08, decay = 0.0

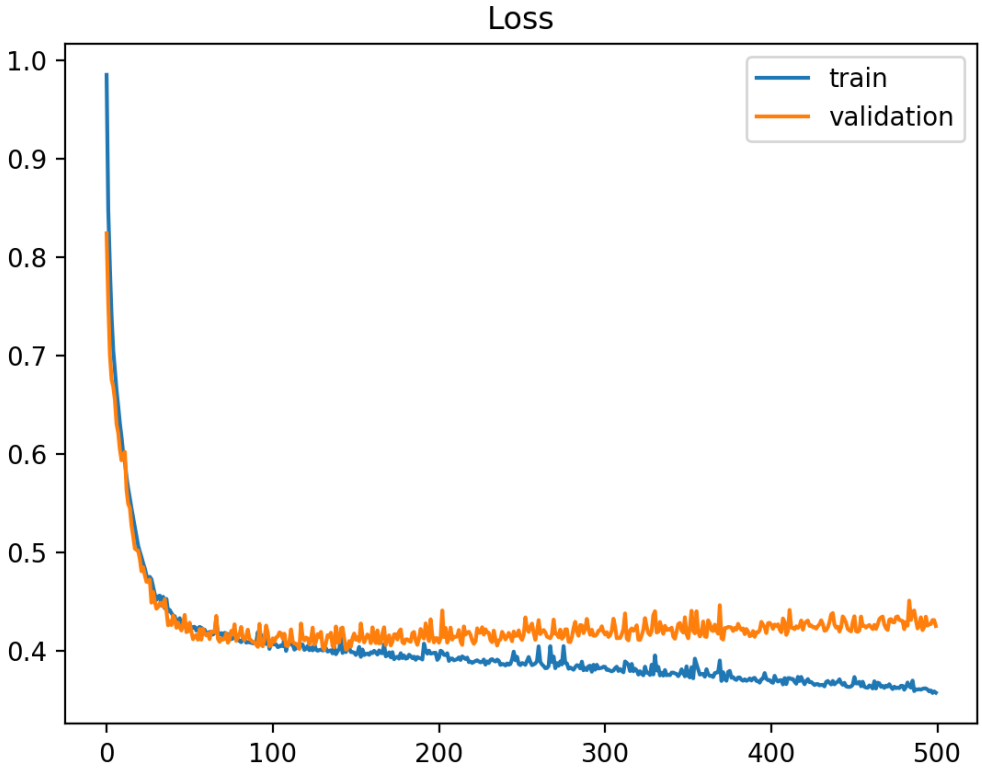
Graph Interprete

1. Underfitting

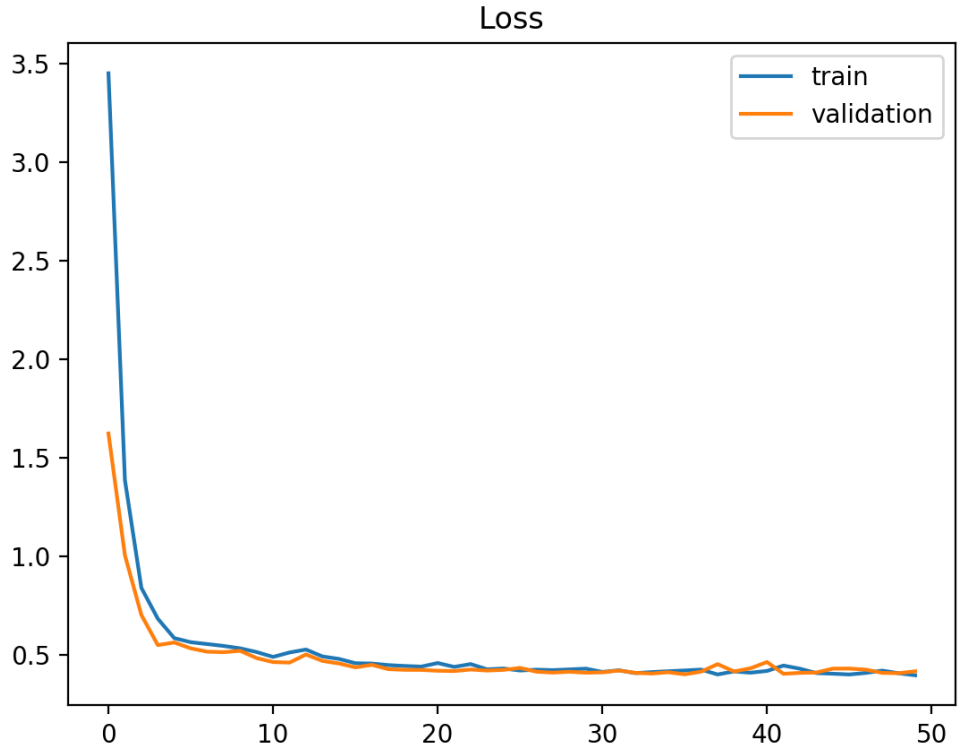




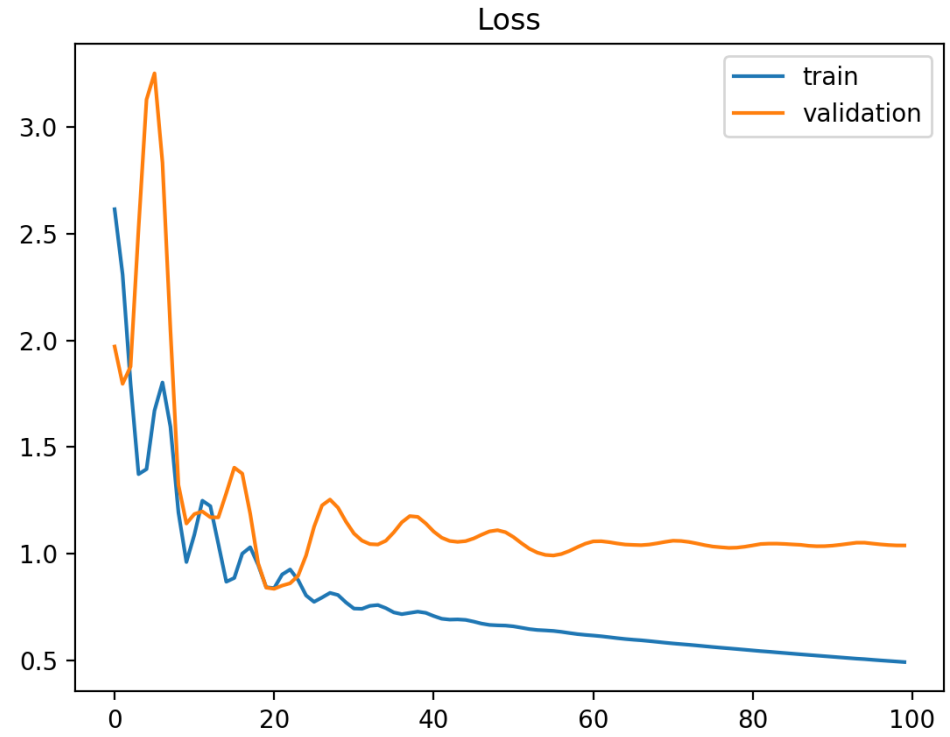
2. Overfitting



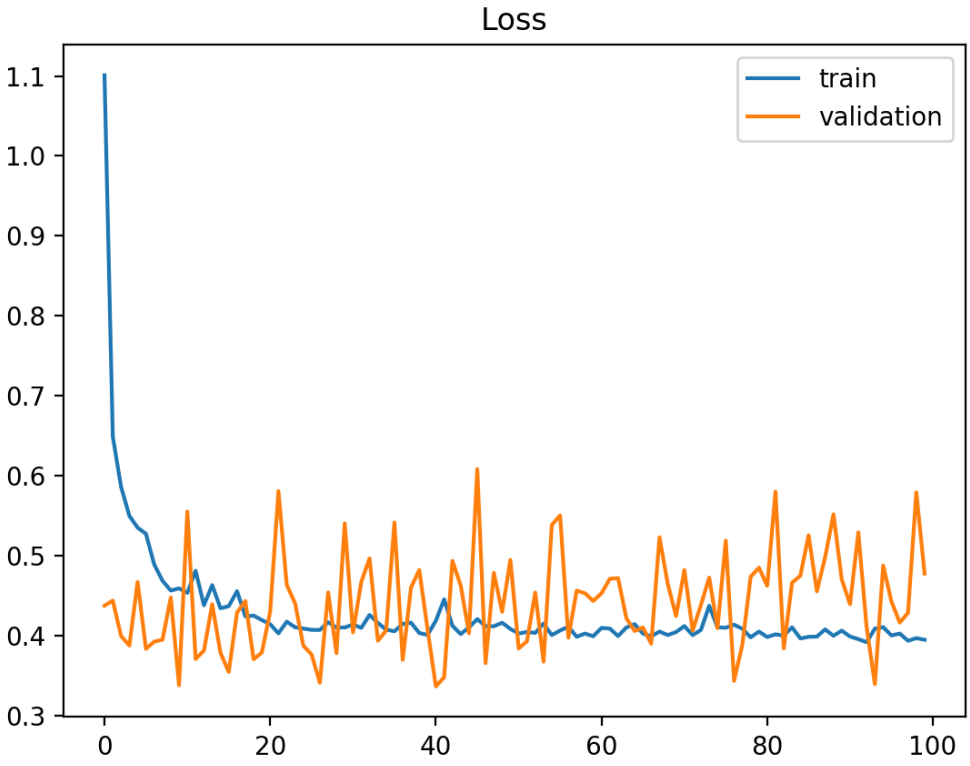
3. Good Fitting

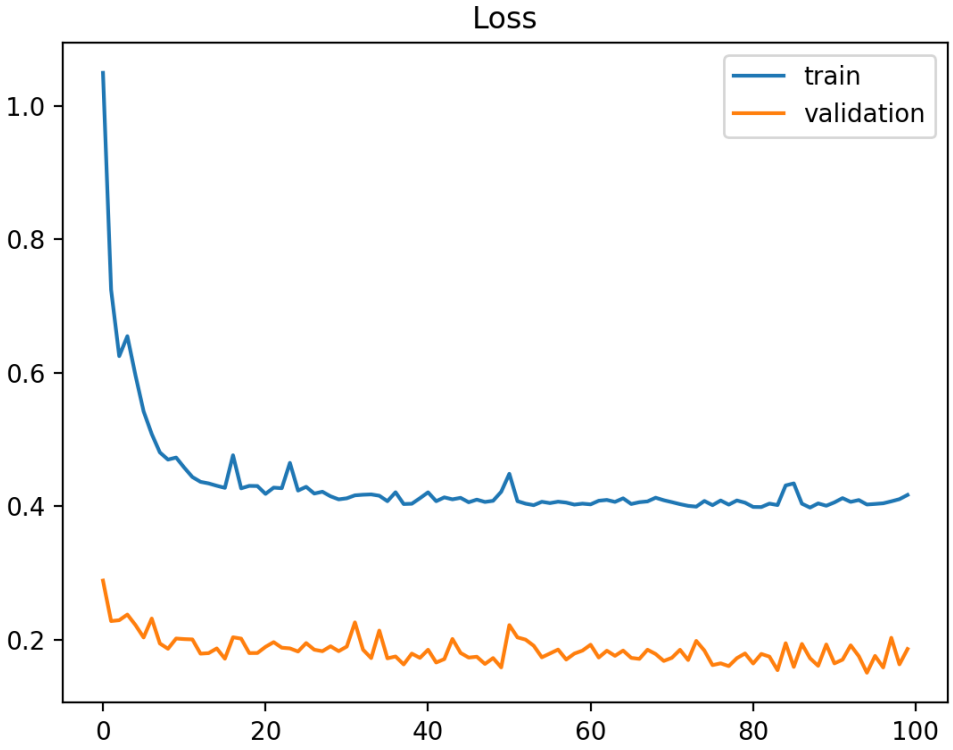


4. a Training Dataset That May Be too Small Relative to the Validation Dataset.



5. a Validation Dataset That May Be too Small Relative to the Training Dataset.




Loss
train
validation
1.0
0.8
0.6
0.4
0.2
0
20
40
60
80
100