یادگیری ماشین

دکتر محمد حسین رهبان

AI CONTEST rayan

مباحث این جلسه

- بهینه سازی(Optimization)
- منظم سازی(Regularization)

بهینه‌سازی

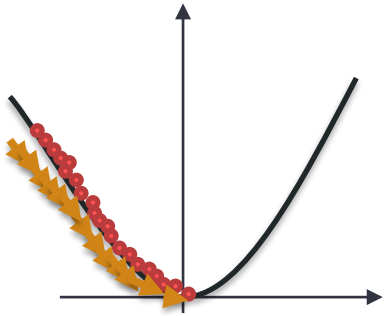
- **یاداوری** :یافتن بهترین وزن ها در شبکه عصبی
- **یاداوری** :الگوریتم گرادیان کاهشی

$$w^{t+1} = w^t - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{\partial loss(y_i, \tilde{y}_i)}{\partial w}$$

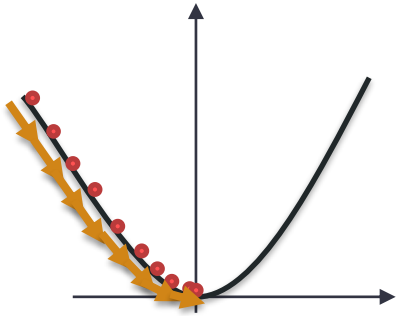
- بررسی روش های مختلف بهینه سازی و مزایا و معایبشان

بهینه‌سازی

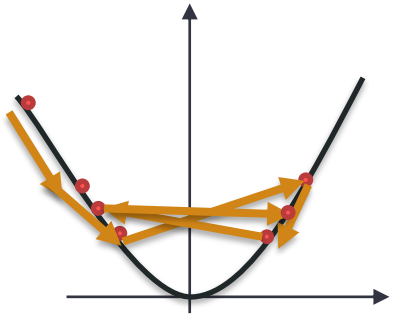
- **تعریف** :کمینه کردن تابع هزینه
- **اهمیت** :یک روش بهینه‌سازی ایده‌آل به ما این اطمینان را میدهد که مدل بطور موثر از داده‌ها استفاده میکند
- **مفهوم کلی** :پیدا کردن یک سیاست و بروزرسانی پارامترها بر اساس آن



❌ بد



✅ خوب



❌ بد

Stochastic Gradient Descent (SGD)

در روش گرادیان کاهشی چه مشکلی وجود دارد؟

- هزینه محاسباتی بسیار زیاد وقتی Nزیاد میشود
- قدرت جست‌جو و بیرون آمدن از مینیمم محلی کم

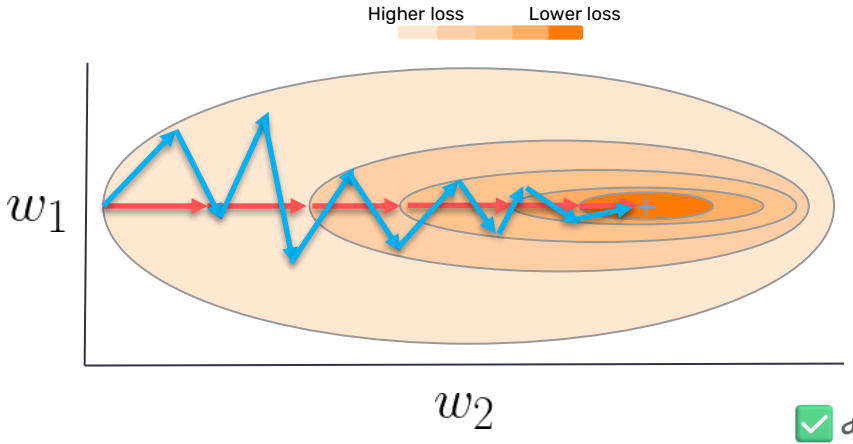
$$w^{t+1} = w^t - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{\partial loss(y_i, \tilde{y}_i)}{\partial w}$$

راه حل :استفاده از قسمت‌های ریزتر یا mini batchداده :(32 / 64 / 128)

$$w^{t+1} = w^t - \alpha \frac{1}{M} \sum_{i=1}^{M} \frac{\partial loss(y_i, \tilde{y}_i)}{\partial w}$$

$$M = mini \;\; batch \;\; size$$

SGD(Stochastic Gradient Descent)



- SGD
- Gradient Descent

گرادیان کاهشی

- سرعت همگرایی بالا ✅
- عدم جست‌وجوی فضای تابع هزینه ❌

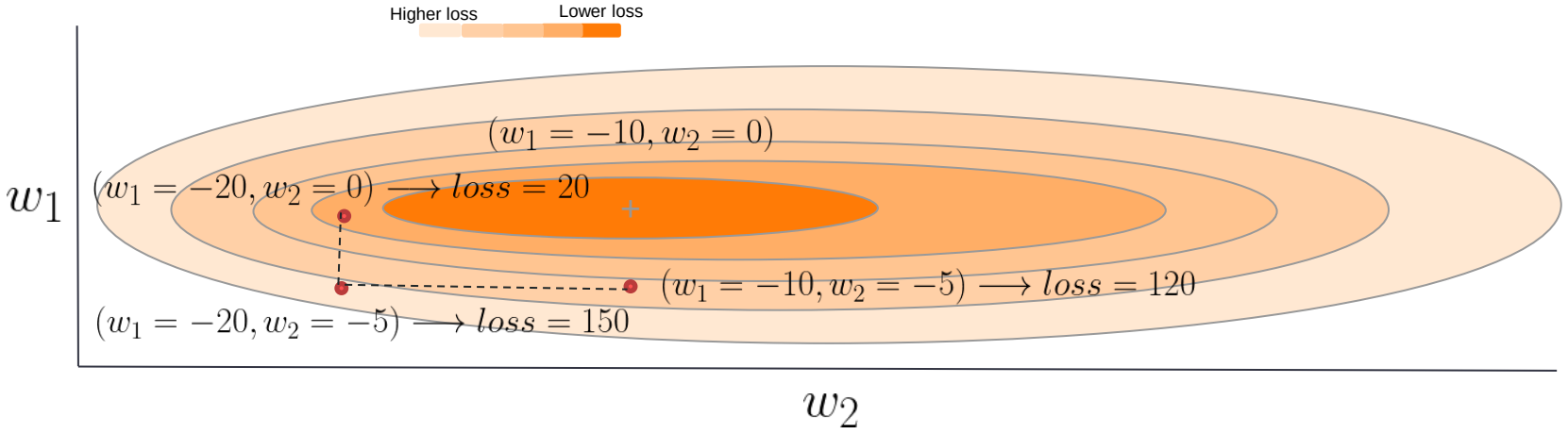
SGD

- سرعت همگرایی کمتر ❌
- جست‌وجوی موثرتر و گسترده‌تر فضای تابع هزینه ✅

مشکلات SGD

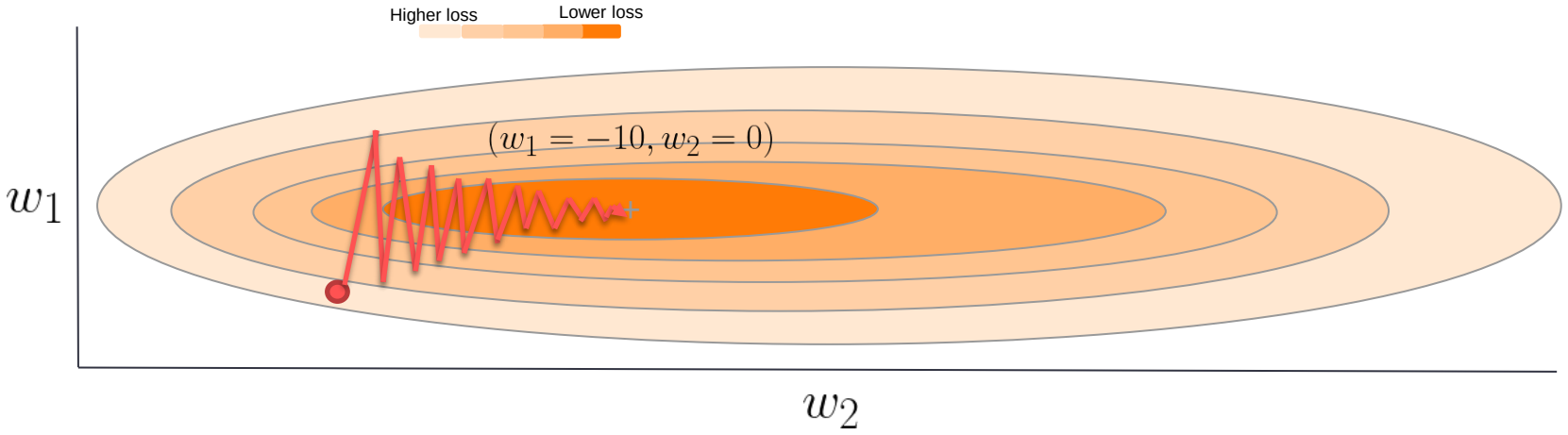
اگر تابع هزینه در یک جهت خیلی بیشتر از یک جهت دیگر تغییر کند؟

عملکرد SGDچگونه خواهد بود؟



مشکلات SGD

پیشرفت در جهت کم عمق بسیار کم و در جهت پر عمق بسیار نوسانی

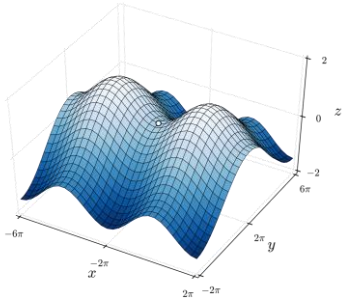


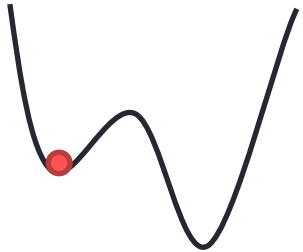
مشکلات SGD

اگر در نقاط زین اسبی یا کمینه محلی باشیم SGDچگونه عمل میکند؟

گرادیان در جهت یا جهت‌هایی صفر میشد و باعث میشود پارامترها در آن جهت‌ها بروز رسانی نشوند

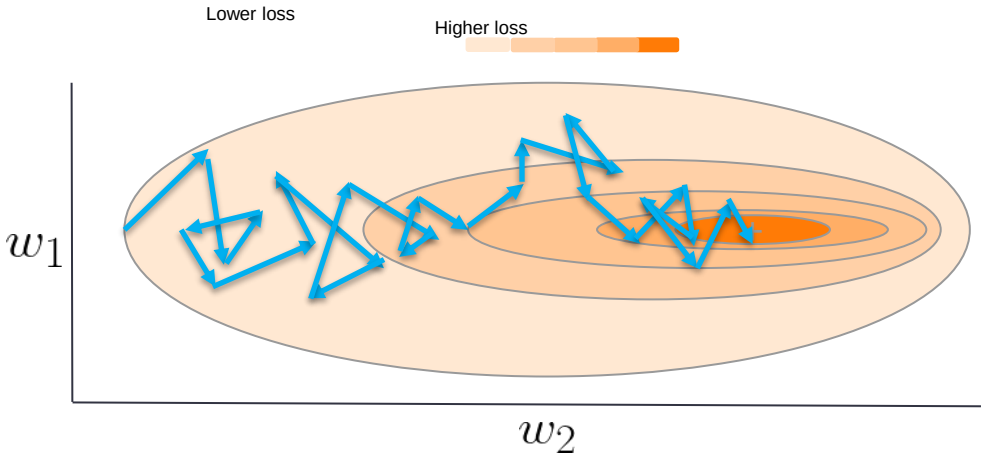
نکته :در شبکه‌های عمیق بیشتر با saddle pointها روبرو هستیم تا مینیمم محلی)چرا؟(





مشکلات SGD

اگر داده‌ها نویزی باشند روند همگرایی نوسانی‌تر نیز میشود چرا که در mini batchها خودشان را بیشتر نشان میدهند

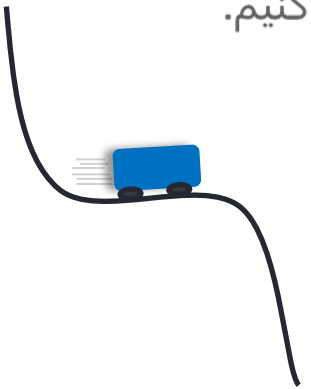


SGD+Momentum

چگونه بر مشکلات SGD غلبه کنیم؟

ایده: بجای بروز رسانی پارامترها بر اساس گرادیان، پارامترها را بر اساس سرعت آپدیت کنیم.

مثل ماشینی در سراشیبی سرعت میگرد و در سطوح صاف مقداری ادامه میدهد.



V مثل سرعت عمل میکند و ρ مثل اصطکاک

SGD+Momentum

در این الگوریتم اگر گرادیان صفر شود چه میشود؟

SGD

$$w^{t+1} = w^t - \alpha \nabla loss(mini\ batch)$$

```
while True:
    dw = compute_gradient(x)
    w = w - alpha * dw
```

SGD + Momentum

$$v^{t+1} = \rho v^t + (1-\rho)\nabla loss(mini\ batch)$$
$$w^{t+1} = w^t - \alpha v^{t+1}$$

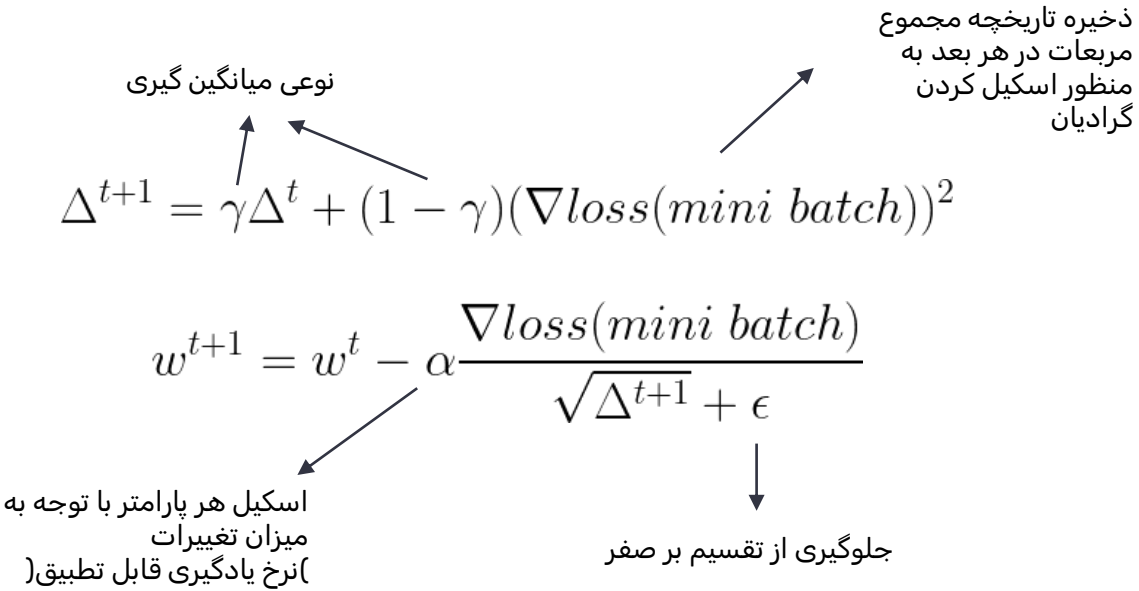
```
v = 0
while True:
    dw = compute_gradient(x)
    v = rho * v + dw
    w = w - alpha * v
```

RMSProp

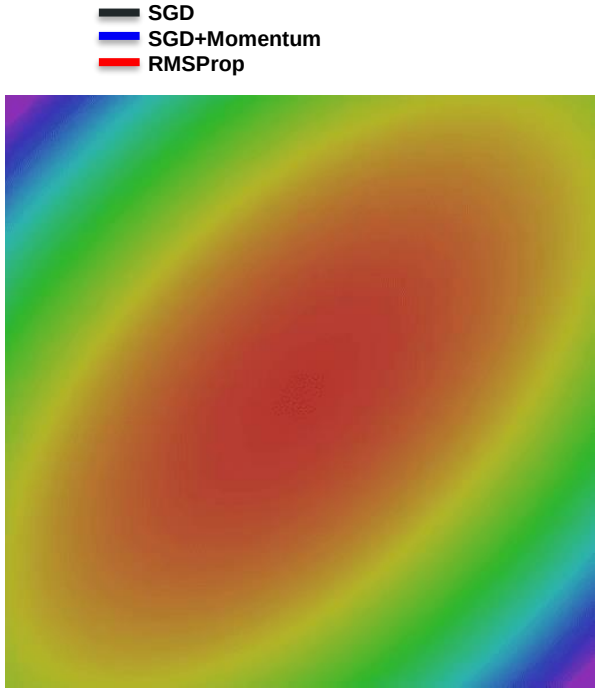
در مورد اختلاف زیاد گرادیان در بعضی از راستا ها چه میشود کرد؟

- برای هر بعد یک نرخ یادگیری قابل تطبیق در نظر بگیریم (adaptive learning-rate)
- پیشرفت در جهاتی که گرادیان زیاد است کند می شود
- پیشرفت در جهاتی که گرادیان کم است شتاب می گیرد

RMSProp


ذخیره تاریخچه مجموع
مربعات در هر بعد به
منظور اسکیل کردن
گرادیان
نوعی میانگین گیری
$\Delta^{t+1} = \gamma\Delta^t + (1-\gamma)(\nabla loss(mini\ batch))^2$
$w^{t+1} = w^t - \alpha\frac{\nabla loss(mini\ batch)}{\sqrt{\Delta^{t+1}} + \epsilon}$
اسکیل هر پارامتر با توجه به
میزان تغییرات
)نرخ یادگیری قابل تطبیق(
جلوگیری از تقسیم بر صفر

RMSProp vs Momentum vs SGD



ADAM

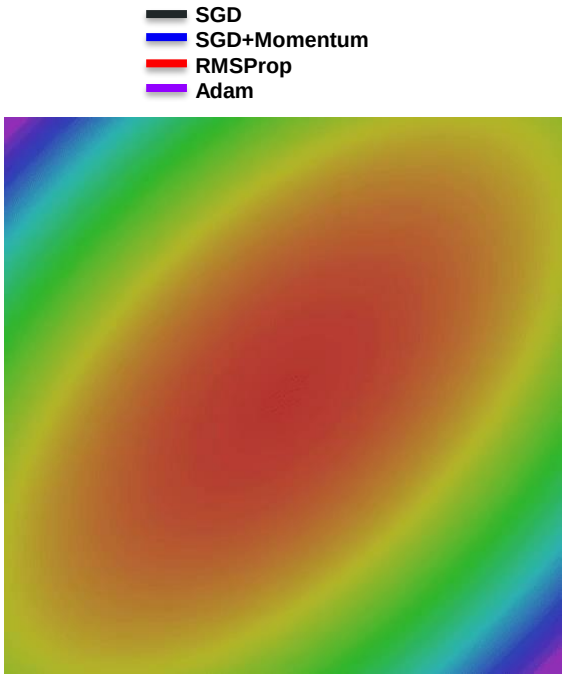
پس بیایم RMSPropو SGD+Momentumرا ترکیب کنیم!

$$\Delta^{t+1} = \gamma_1 \Delta^t + (1-\gamma_1)(\nabla loss(mini\ batch))^2$$ **Momentum**

$$v^{t+1} = \gamma_2 v^t + (1-\gamma_2)\nabla loss(mini\ batch)$$ **RMSProp**

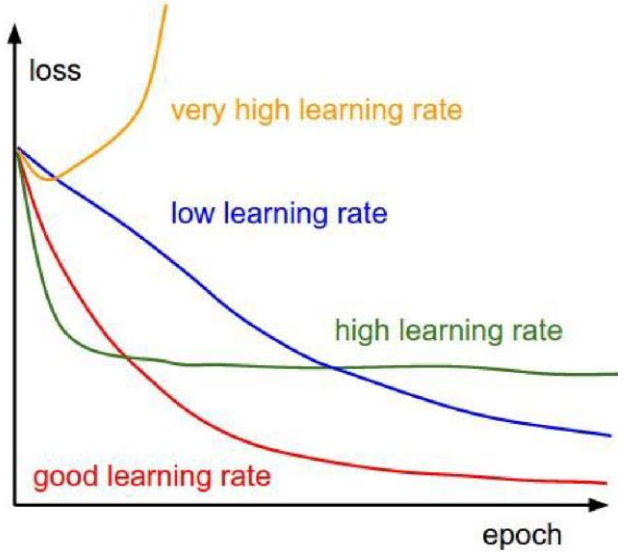
$$w^{t+1} = w^t - \alpha \frac{v^{t+1}}{\sqrt{\Delta^{t+1}} + \epsilon}$$ **RMSProp + Momentum**

ADAM



Learning rates

انتخاب Optimizerیا بهینه‌ساز با توجه به نوع مساله و آزمون و خطا بدست میاید ولی در موارد بسیار Adamگزینه خوبی برای انتخاب است.

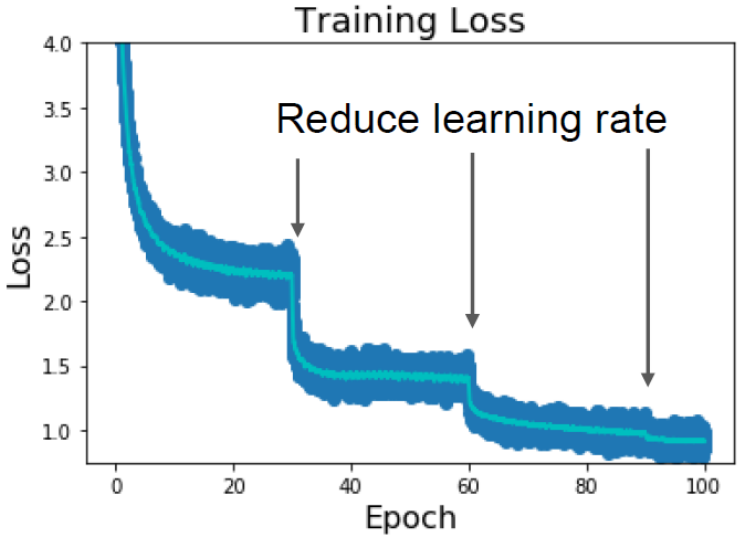


آیا انتخاب بهینه‌ساز به تنهایی کافی است؟

Learning rates

چه استراتژی‌هایی را برای انتخاب نرخ یادگیری می توان به کار برد؟

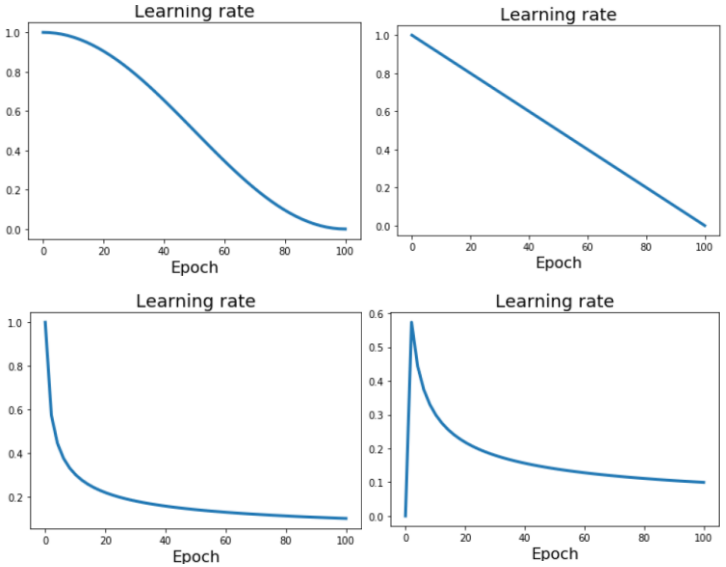
- کاهش نرخ یادگیری در هر چند step



Learning rates

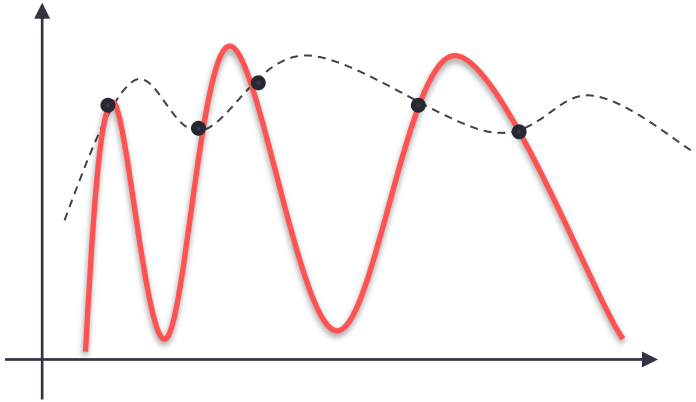
چه استراتژی‌هایی را برای انتخاب نرخ یادگیری می توان به کار برد؟

- استفاده از توابع پیچیده‌تر برای کاهش نرخ یادگیری



Regularization

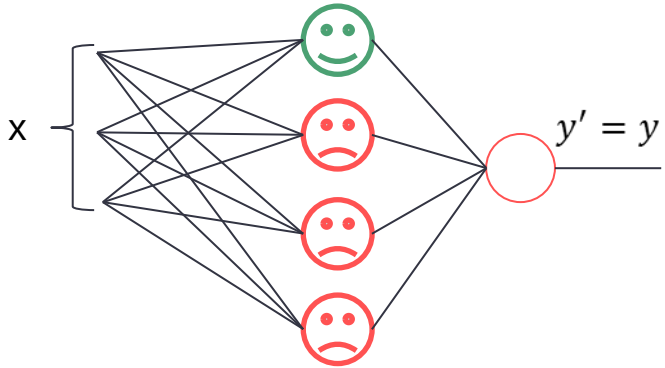
مشکل :شبکه ممکن است که فقط مقادیر ورودی در هنگام آموزش را یاد بگیرد

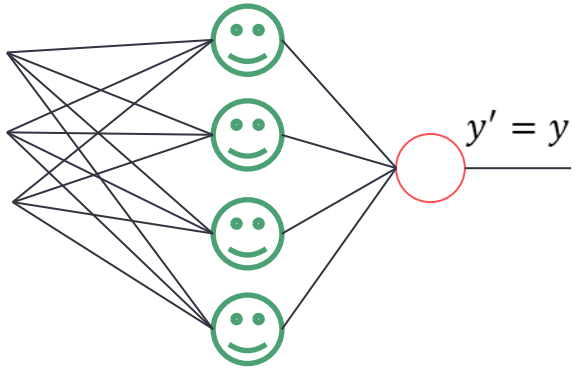


Regularization

هدف: جلوگیری از overfitشدن مدل و بهبود تعمیم پذیری (generalization)مدل

اهمیت: دنبال مدل‌هایی هستیم که رو داده دیده نشده خوب عمل کنند و تکنیک‌های regularizationبه ما کمک میکنند تا به این هدف دست پیدا کنیم





Regularization

دو دسته روش برای منظم سازی میتوان در نظر گرفت

ضمنی

Implicit

صریح

Explicit

Regularization (Explicit)

تعریف :اضافه کردن یک عبارت جدید به تابع هزینه به منظور وارد کردن Inductive Bias مورد نظر

- L1 Regularization
- L2 Regularization

Regularization (Explicit)

یاداوری از تابع هزینه:

$$L = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i, W), y_i)$$

هزینه داده:
پیشبینی مدل باید با لیبل همخوانی داشته باشد

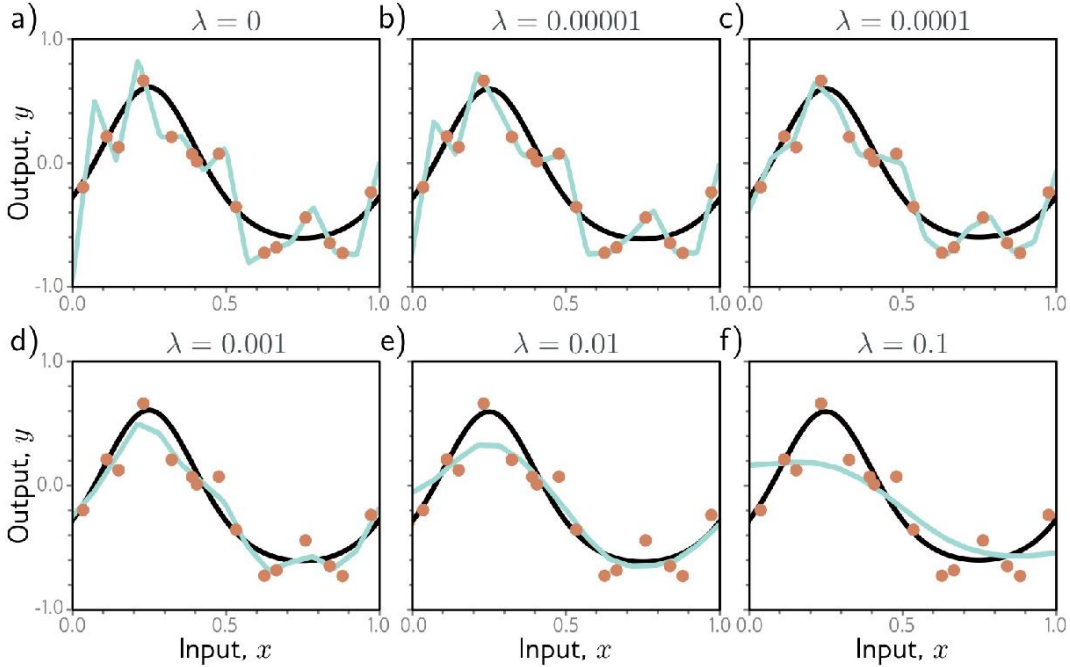
در تنظیم‌ساز های ابتدایی یک عبارت برحسب پارامترها اضافه میشود به منظور کنترل و مقید کردن پارامترها.

$$L = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i, W), y_i) + \lambda R(W)$$

λ: ضریب منظم ساز

منظم ساز:
جلوگیری از اینکه مدل روی داده‌ی آموزش بیش از اندازه خوب عمل کند .

Regularization (Explicit)



تاثیر لاندا λ:

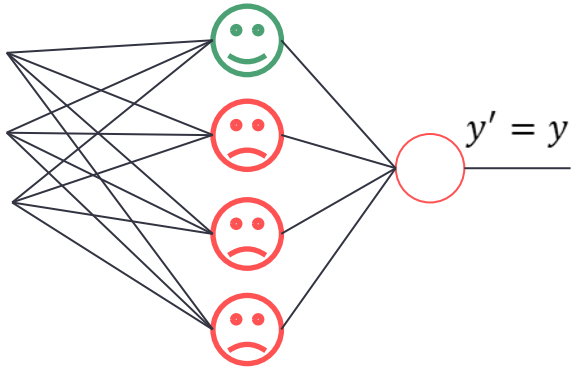
L1 Regularization

$$L = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i, W), y_i) + \lambda R(W)$$

چند مثال ساده از منظم سازها:

- L1: $R(W) = \sum_k \sum_l |W_{k,l}|$

تشویق مدل به صفر کردن پارامترها(sparsity) (چرا؟)



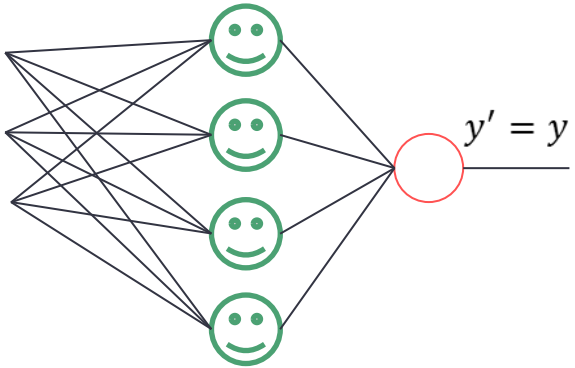
L2 Regularization

$$L = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i, W), y_i) + \lambda R(W)$$

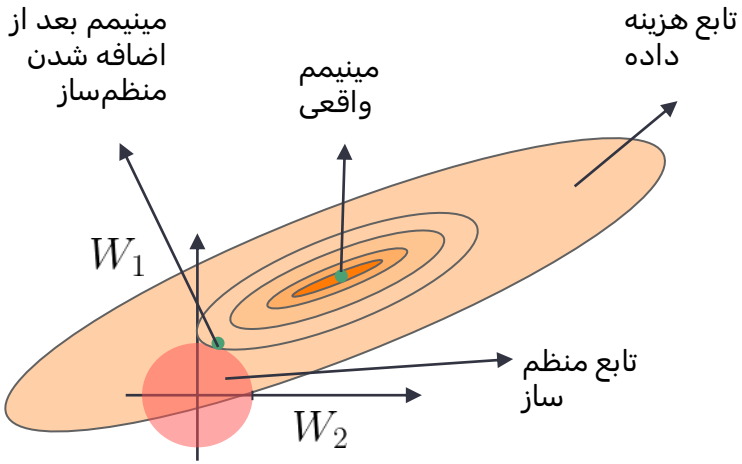
چند مثال ساده از منظم سازها:

- L2: $R(W) = \sum_k \sum_l W_{k,l}^2$

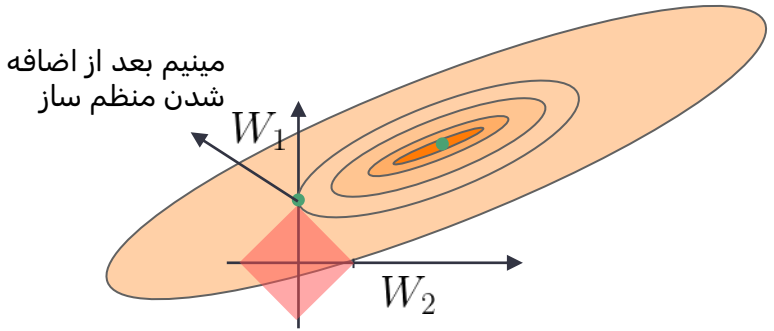
تشویق مدل به پخش کردن مقادیر بین پارامترها (چرا؟)



Regularization (Explicit)

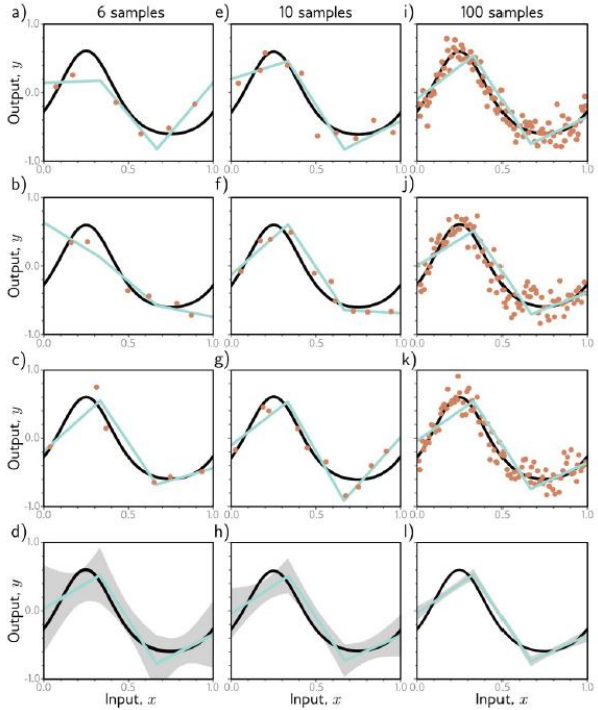


$$R(W) = W_1^2 + W_2^2 = T^2$$



$$R(W) = |W_1| + |W_2| = T$$

Regularization (Implicit)



تاثیر تعداد داده‌ها در واریانس مدل‌ها

چرا واریانس کم بین مدل‌های ارائه شده مطلوب است؟

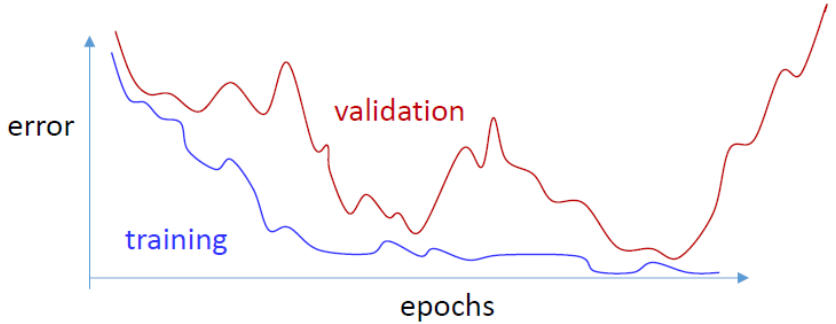
Regularization (Implicit)

روش های رایج منظم سازی ضمنی

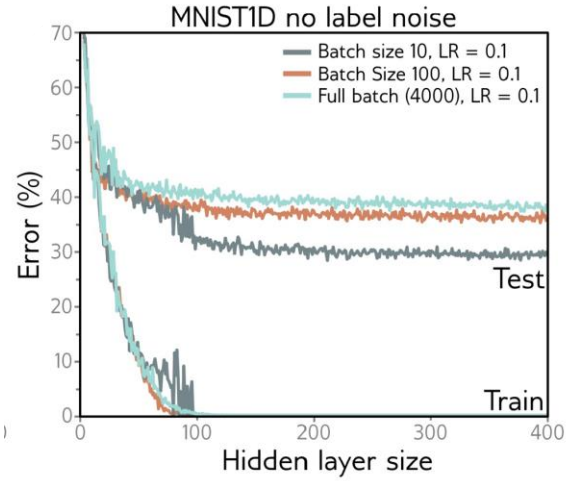
- افزایش داده
- Dropout
- Data augmentation
- توقف زود هنگام (early stopping)
- روش‌های Ensemble

Early Stopping

در این روش تحت شرایطی آموزش مدل متوقف میشود چرا که از یک زمان به بعد خطای داده ارزیابی دیگر بهبود چشم‌گیری پیدا نمیکند



Regularization (Implicit)



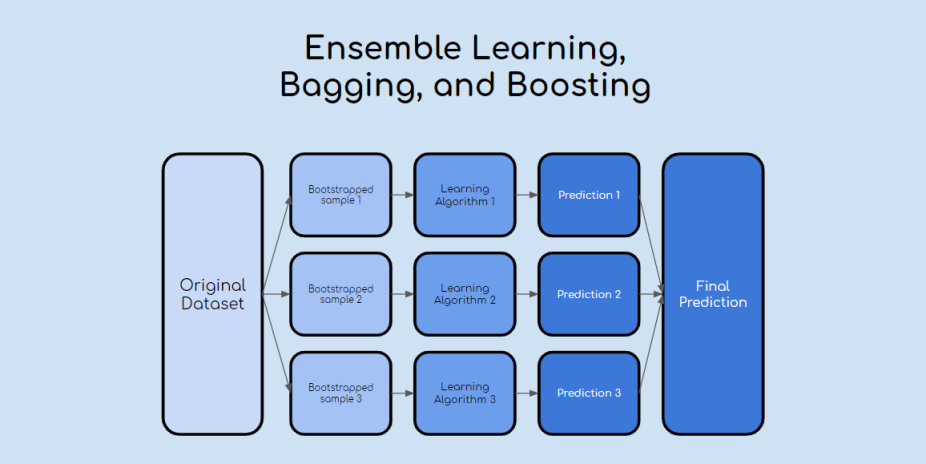
SGDبهتر از گرادیان کاهشی تعمیم پذیری دارد:

- بچ‌سایز کوچکتر بهتر از بچ‌سایز بزرگ تعمیم پذیری دارد

- وجود randomnessذاتی در بچ‌ها به مدل این اجازه را میدهد که فضای گسترده‌تری از تابع هزینه را جست‌وجو کند

Ensemble Methods

در این روش به جای استفاده از یک مدل، چند مدل آموزش می بینند و در پیشبینی هنگام تست از تجمیع پیشبینی مدل ها استفاده می کند



Ensemble Methods

چگونه چند مدل آموزش بدهیم

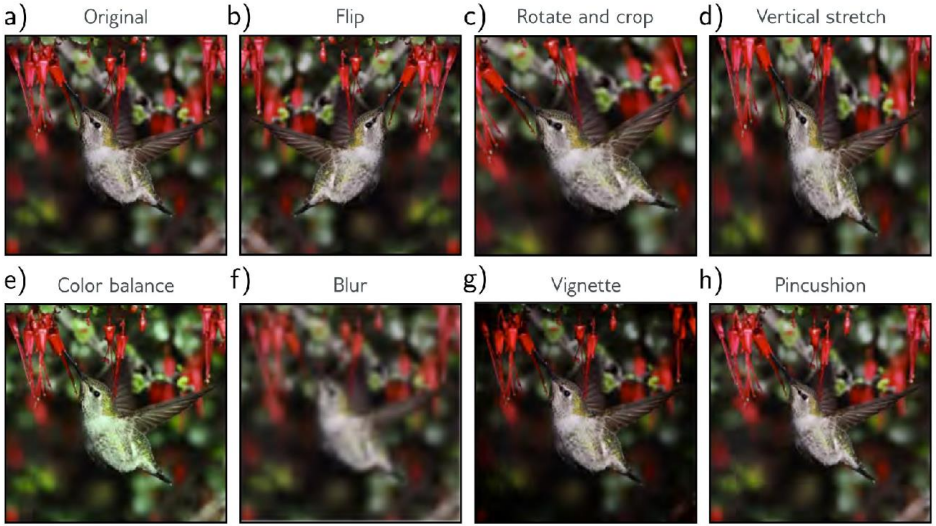
- آموزش چند مدل با قسمت‌های مختلف داده یا نقاط شروع مختلف
- استفاده از بهترین مدل‌ها بعد از cross-validation
- استفاده از چند سری از وزن‌های ذخیره شده به عنوان مدل‌ها

روش‌های تجمیع پیشبینی‌ها

- رای اکثریت
- بیشترین اطمینان (Highest confidence)
- جمع احتمالات

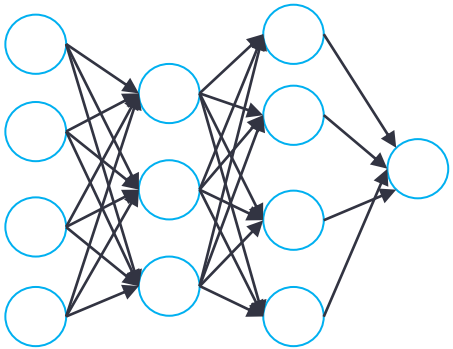
Data Augmentation

- در بعضی مواقع جمع‌آوری داده بسیار هزینه‌بر است
- ساختن داده مصنوعی بطوریکه برچسب آن با داده اصلی یکسان باشد

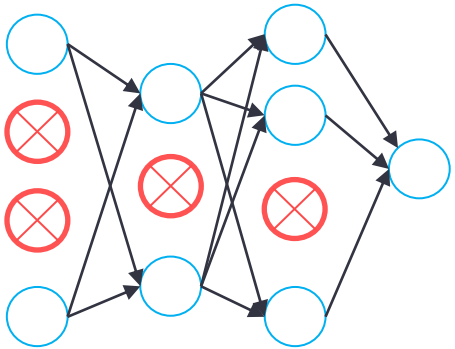


Dropout Method

- خاموش کردن تصادفی بعضی نورون‌ها هنگام آموزش مدل
- جلوگیری از وابستگی مدل به یکسری نورون‌های خاص



شبکه عصبی



شبکه عصبی پس از اعمال Dropout

References

- https://medium.datadriveninvestor.com
- https://artemoppermann.com/
- Understanding Deep Learning, 2023