

آموزش یادگیری عمیق

Deep Learning

« آموزش شبکه‌های عصبی »

سعید محقق / دانشگاه شاهد / ۹۹ - ۱۳۹۸

آموزش شبکه‌های عصبی

- ۱- آماده سازی داده‌های آموزش
- ۲- انتخاب معماری مناسب شبکه
- ۳- الگوریتم آموزش و بهینه‌سازی
- ۴- روش‌های بهبود آموزش

اصول آماده سازی داده‌ها

- داده بیشتر = آموزش بهتر
- حذف داده‌های ناقص و مخدوش
- پیش پردازش داده‌ها (یکسان سازی شرایط)
- استفاده از روش‌های افزایش داده (Data augmentation)

پیش پردازش داده‌ها

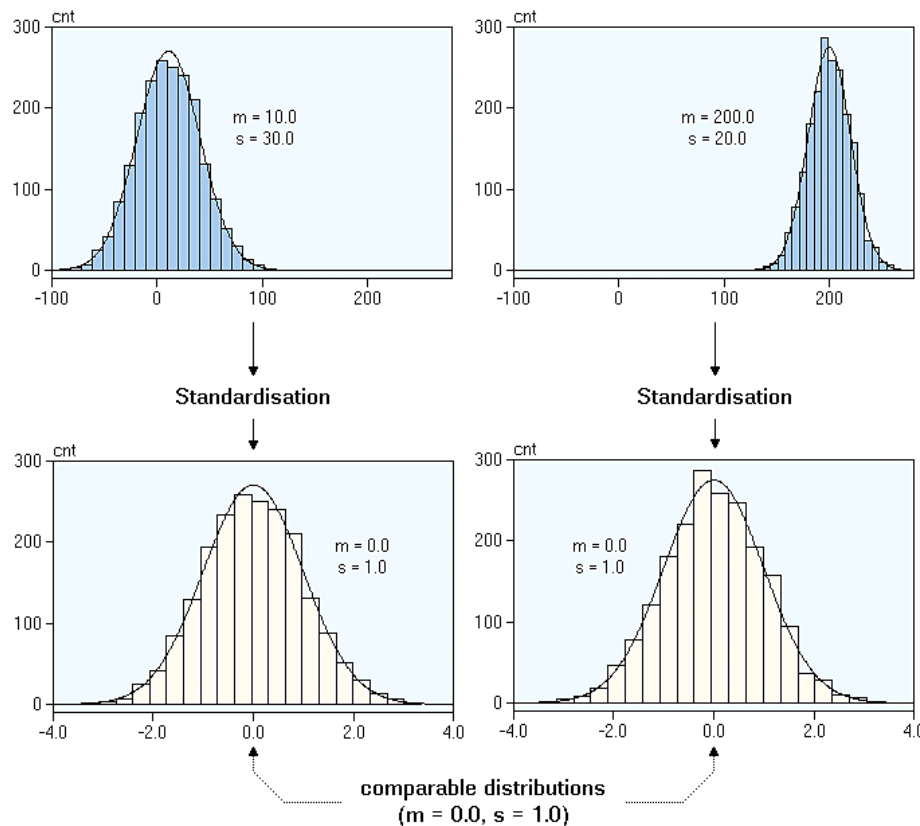
■ هدف

- حذف اطلاعات و تغییرات نامطلوب و مشابه سازی داده‌ها

■ مثال

- حذف تغییرات نور و یکسان سازی محدوده شدت روشنایی تصاویر
- یکسان سازی مقیاس در داده‌های عددی

پیش پردازش داده‌ها



■ استاندارد کردن

■ تغییر داده‌ها برای داشتن

میانگین 0 و واریانس 1

$$x_{\text{standardized}} = \frac{(x - \text{mean}(x))}{\text{std}(x)}$$

پیش پردازش داده‌ها

■ نرمالیزه کردن

■ انتقال محدوده مقادیر داده‌ها به 0 و 1

$$x_{normalized} = \frac{(x - \min(x))}{(\max(x) - \min(x))}$$

پیش پردازش داده‌ها

■ یکسان سازی هیستوگرام تصاویر (Histogram equalization)



Before Histogram Equalization



After Histogram Equalization

افزایش داده‌ها

■ ایجاد تغییرات بر روی داده‌های فعلی و ایجاد داده‌های جدید

■ مثال: جابجایی / تغییر اندازه / چرخش / تغییر شکل و رنگ



انتخاب معماری شبکه

■ تعداد لایه‌ها و نورون‌ها (فیلترها)

1. استفاده از معماری مدل‌های آماده
2. شروع با تعداد کم ← افزایش، تا جایی که نتیجه بهبود پیدا نکند. (یا برعکس!)
3. الگوریتم‌های جستجوی معماری (Neural Architecture Search)

انتخاب معماری شبکه

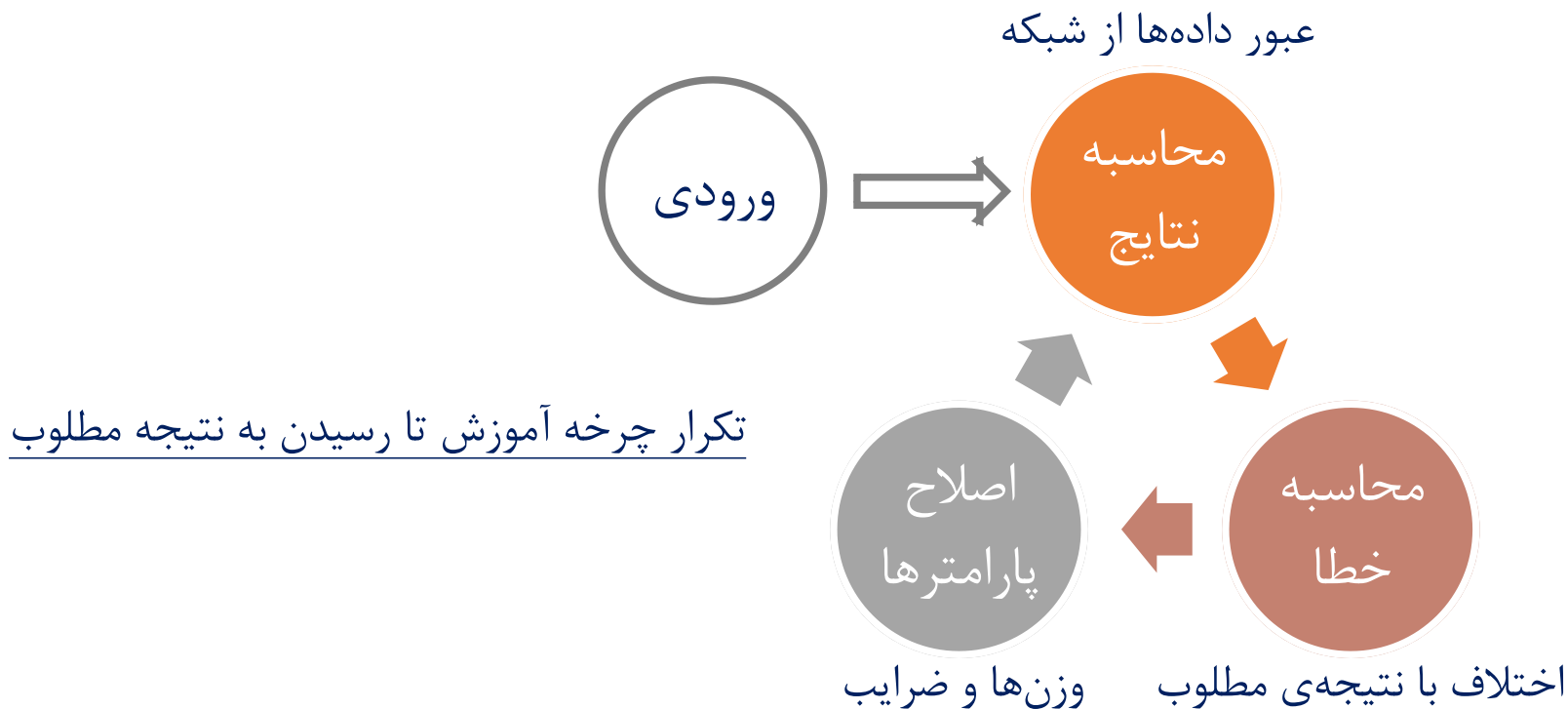
■ نوع شبکه

■ داده یک بعدی (سیگنال، بردار) ← شبکه FC یا AE

■ داده چندبعدی (تصویر، تنسور) ← شبکه CNN

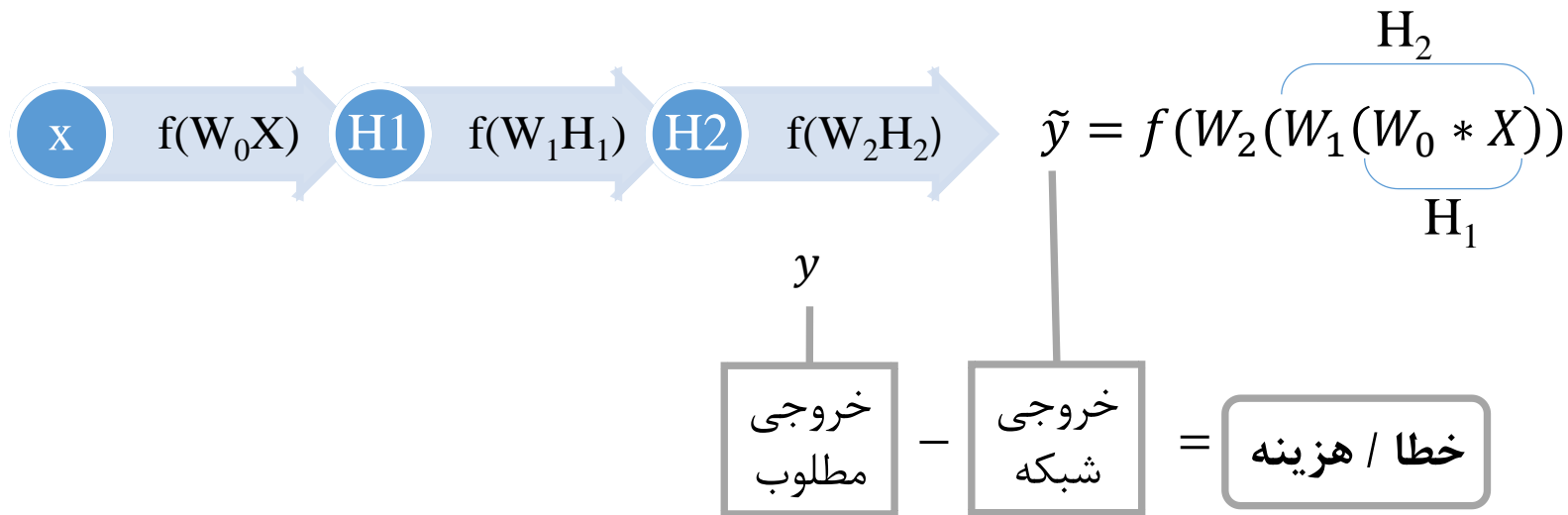
■ داده وابسته به زمان (صوت، ویدیو، متن) ← شبکه RNN

آموزش شبکه



محاسبه خطا

■ عبور داده‌ها از شبکه و مقایسه با نتیجه مطلوب



محاسبه خطا

■ انتخاب تابع هزینه = انتخاب نحوه محاسبه خطا

■ نمونه‌های تابع هزینه (Cost / Loss function)

Mean squared –

Cross-entropy –

Hinge –

Dice –

Mean squared:
$$L(y, \tilde{y}) = \frac{1}{N} \sum_i (y_i - \tilde{y}_i)^2$$

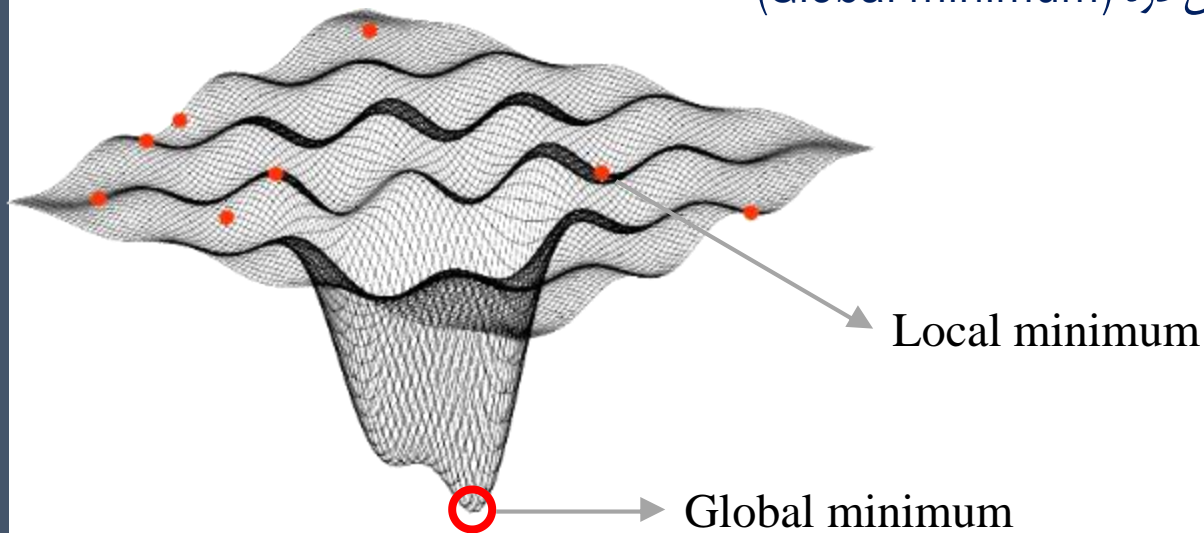
بهینه‌سازی (Optimization)

- حرکت قدم به قدم به سمت کم‌ترین مقدار خطا
- اصلاح و به‌روزرسانی وزن‌ها برای رسیدن به حداقل خطا

بهینه‌سازی (Optimization)

■ مقادیر مختلف تابع هزینه در حالت های مختلف ← تشکیل یک سطح ناهموار (Loss surface)

■ هدف ← رسیدن به عمیق ترین دره (Global minimum)



الگوریتم بهینه‌سازی

- آموزش کامل:

<http://ruder.io/optimizing-gradient-descent/>

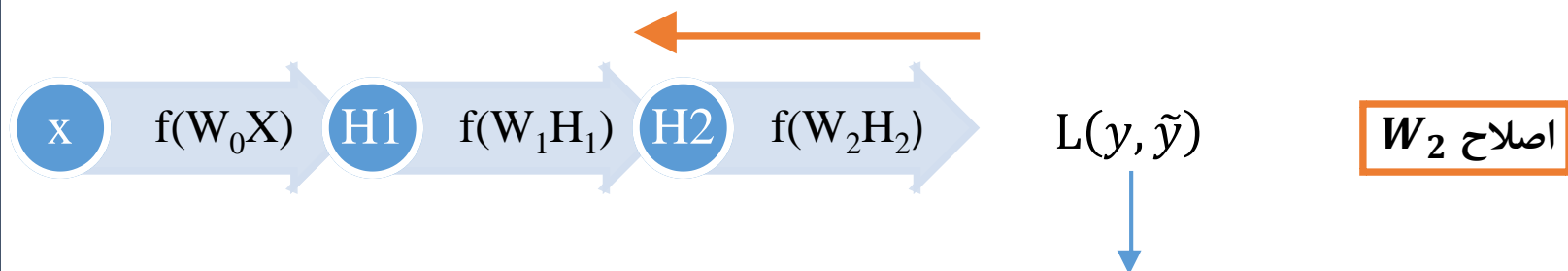


■ الگوریتم‌ها

- SGD -
- SGD + Momentum -
- RMSprop -
- Adagrad -
- Adadelat -
- Adam -
- ...

اصلاح وزن‌ها

■ روش Back-propagation



$$\Delta W_2 = \frac{\partial L}{\partial W_2}$$

- محاسبه گرادیان L نسبت به وزن W_2 :

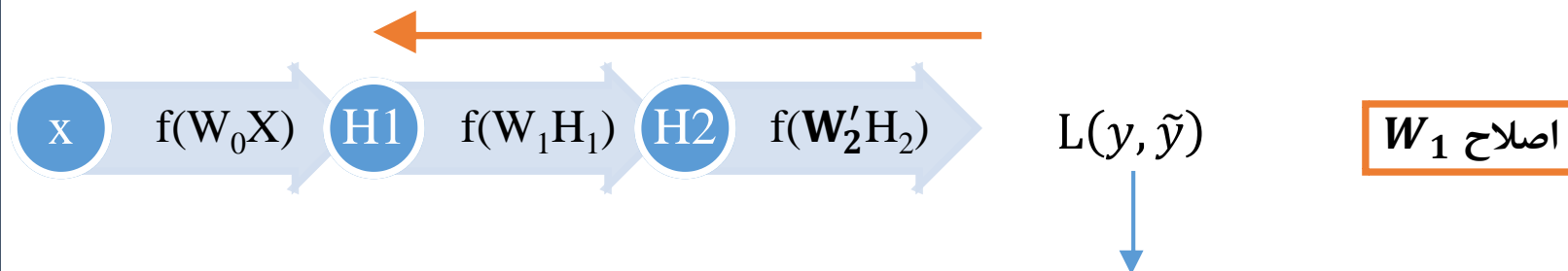
$$W_2' = W_2 - \alpha(\Delta W_2)$$

- اصلاح W_2 در جهت کاهش گرادیان L (الگوریتم SGD) :

↓
Learning rate

اصلاح وزن‌ها

■ روش Back-propagation



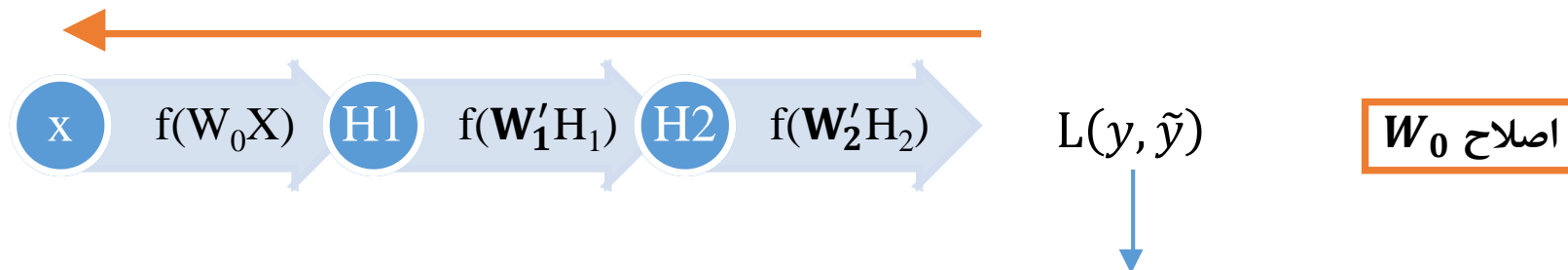
$$\Delta W_1 = \frac{\partial L}{\partial W_1}$$

$$W'_1 = W_1 - \alpha(\Delta W_1)$$

- محاسبه گرادیان L نسبت به وزن W_1 :
- اصلاح W_1 در جهت کاهش گرادیان L :

اصلاح وزن‌ها

■ روش Back-propagation



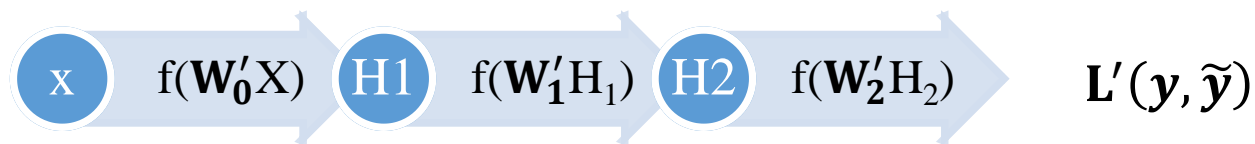
$$\Delta W_0 = \frac{\partial L}{\partial W_0}$$

$$W'_0 = W_0 - \alpha(\Delta W_0)$$

- محاسبه گرادیان L نسبت به وزن W_0 :
- اصلاح W_0 در جهت کاهش گرادیان L :

اصلاح وزن ها

■ روش Back-propagation



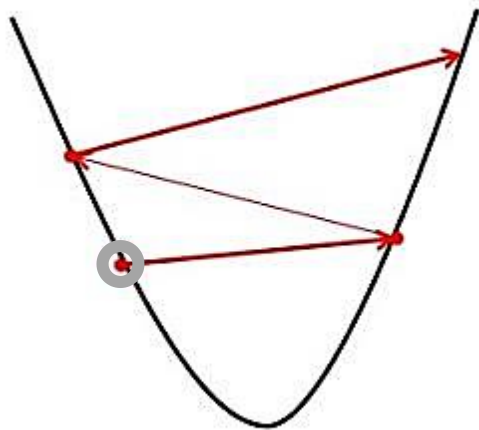
محاسبه مجدد خطا

تکرار اصلاح وزن ها با خطای جدید

نرخ آموزش

■ اندازه گامها به سمت حداقل خطا

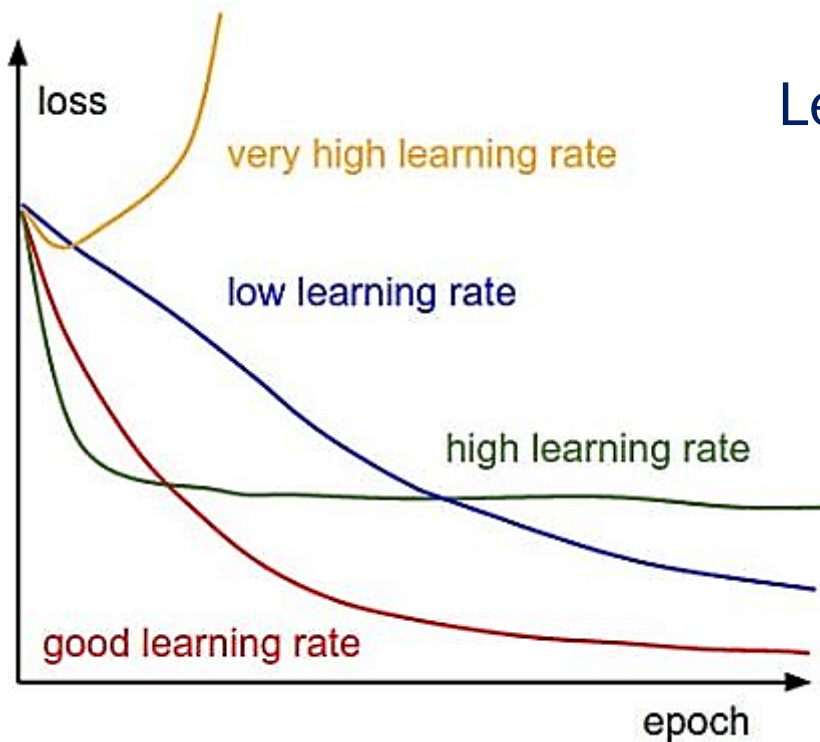
Big learning rate



Small learning rate



نرخ آموزش



■ اثر مقادیر مختلف Learning rate

روش‌های بهبود آموزش

Batch normalization ■

Dropout ■

Transfer learning ■

Batch Normalization

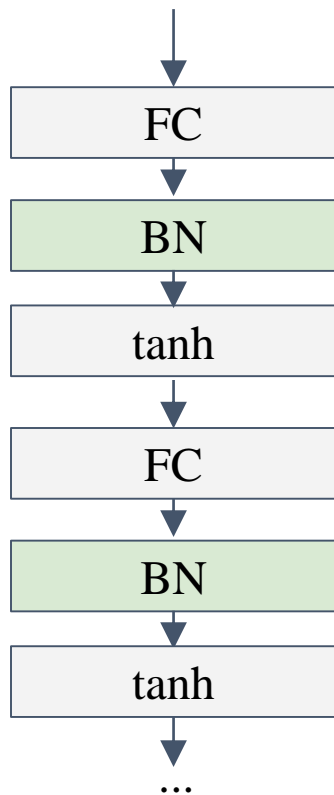
■ مزایا

■ حفظ مقادیر خروجی لایه‌ها در محدوده نرمال

■ جلوگیری از over-fitting

■ افزایش سرعت و دقت آموزش

Batch Normalization



به عنوان یک لایه
بعد از اعمال وزن ها و قبل از تابع غیرخطی

$$\hat{x}^{(k)} = \frac{x^{(k)} - \text{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}}$$

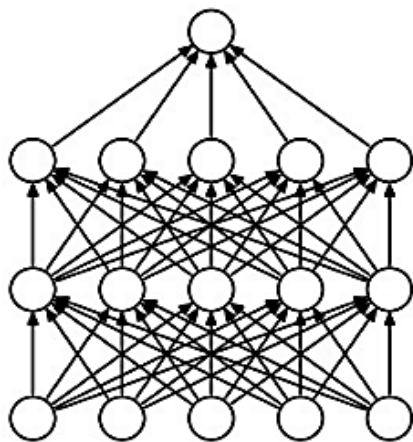
Dropout

■ حذف تصادفی تعدادی از نورون‌ها در هر تکرار آموزش

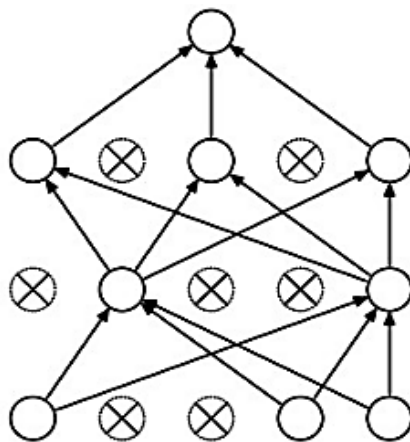
■ مزایا:

- آموزش مستقل نورون‌ها

- جلوگیری از Over-fitting



(a) Standard Neural Net



(b) After applying dropout.

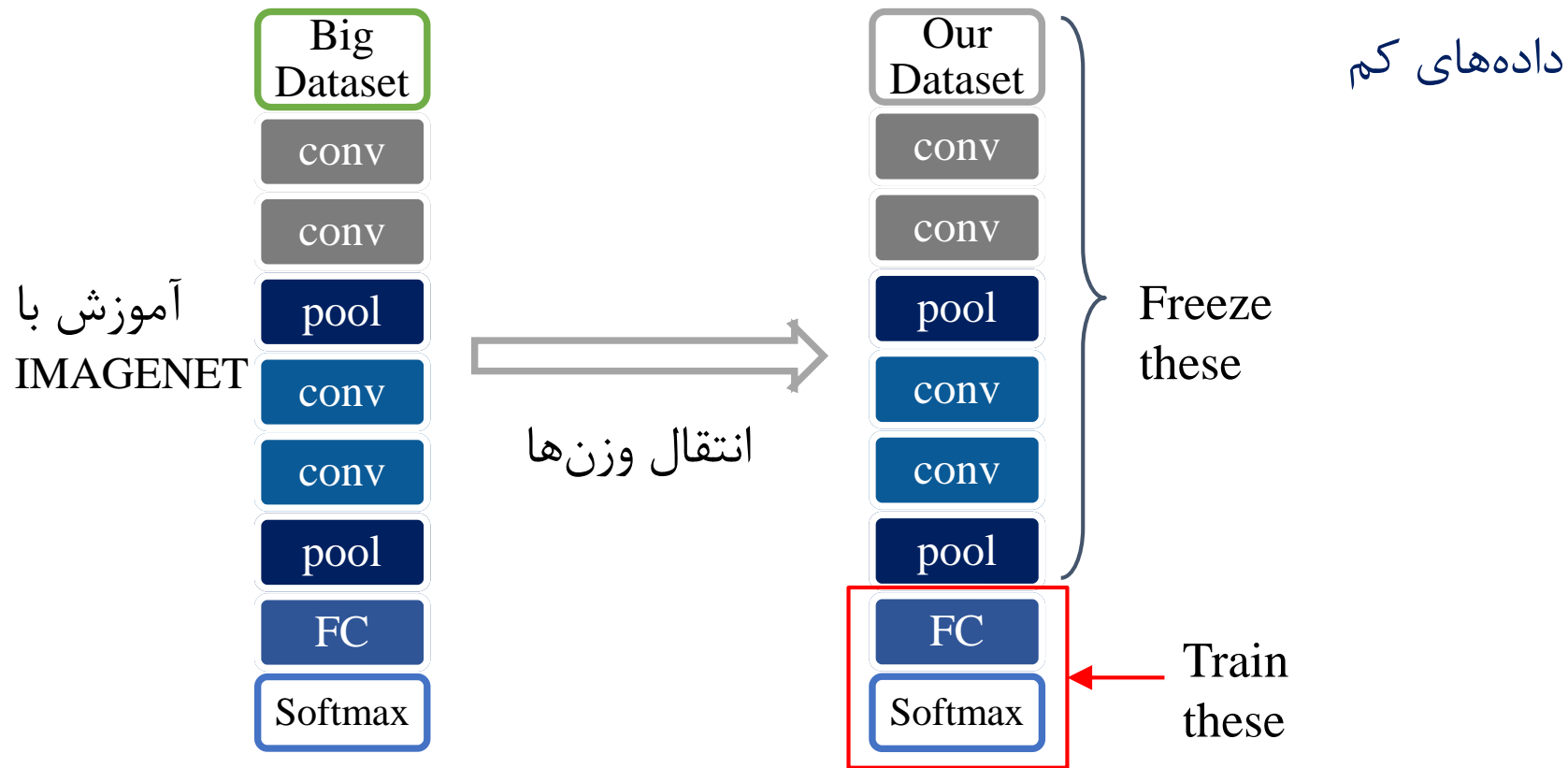
انتقال یادگیری (Transfer Learning)

■ رفع مشکل کمبود داده‌های آموزش

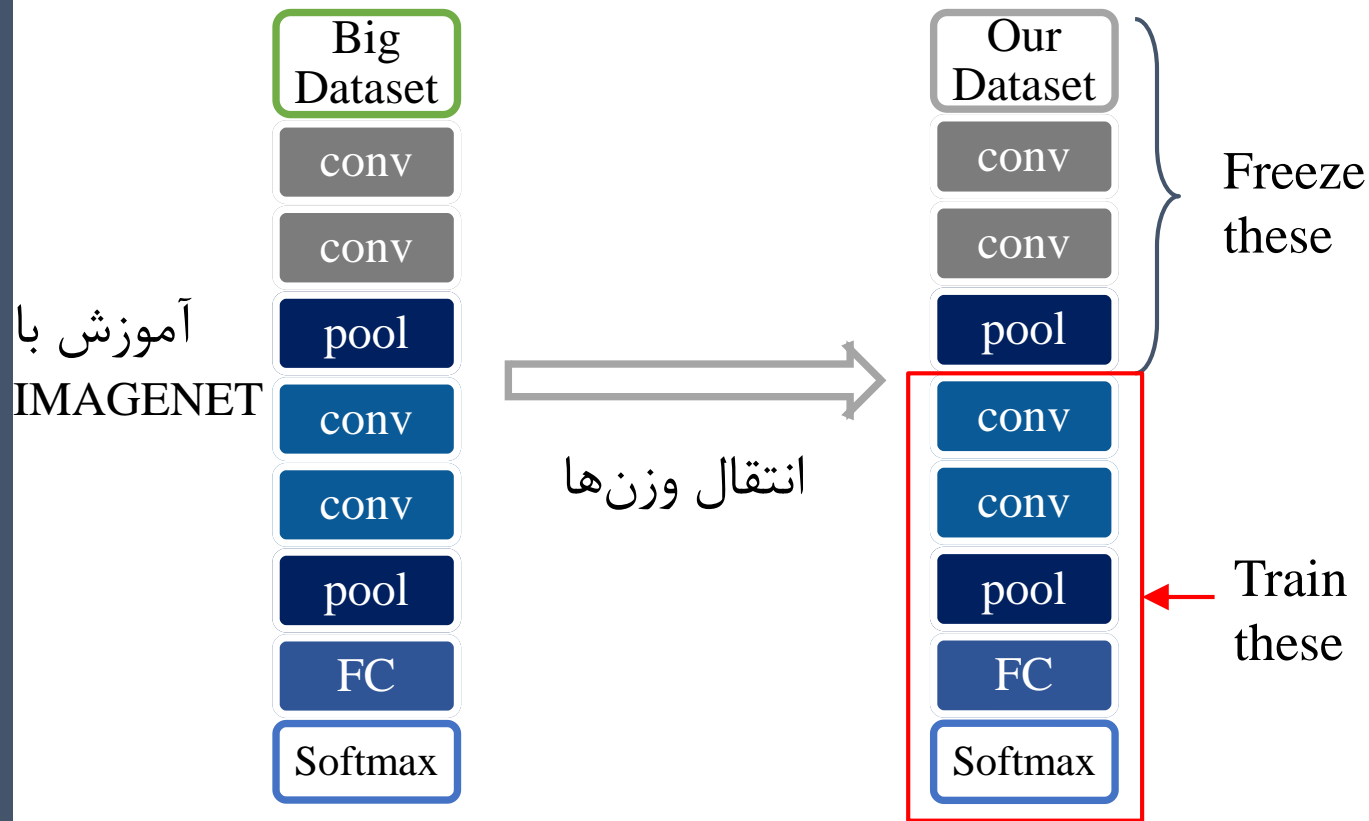
■ مراحل کار:

1. آموزش شبکه با یک دیتاست بزرگ
2. انتقال وزن‌های شبکه آموزش دیده به مدل جدید
3. اصلاح جزئی آموزش با داده‌های مورد نظر

انتقال یادگیری (Transfer Learning)



انتقال یادگیری (Transfer Learning)



داده های متوسط

پایان