

رسالة محمد

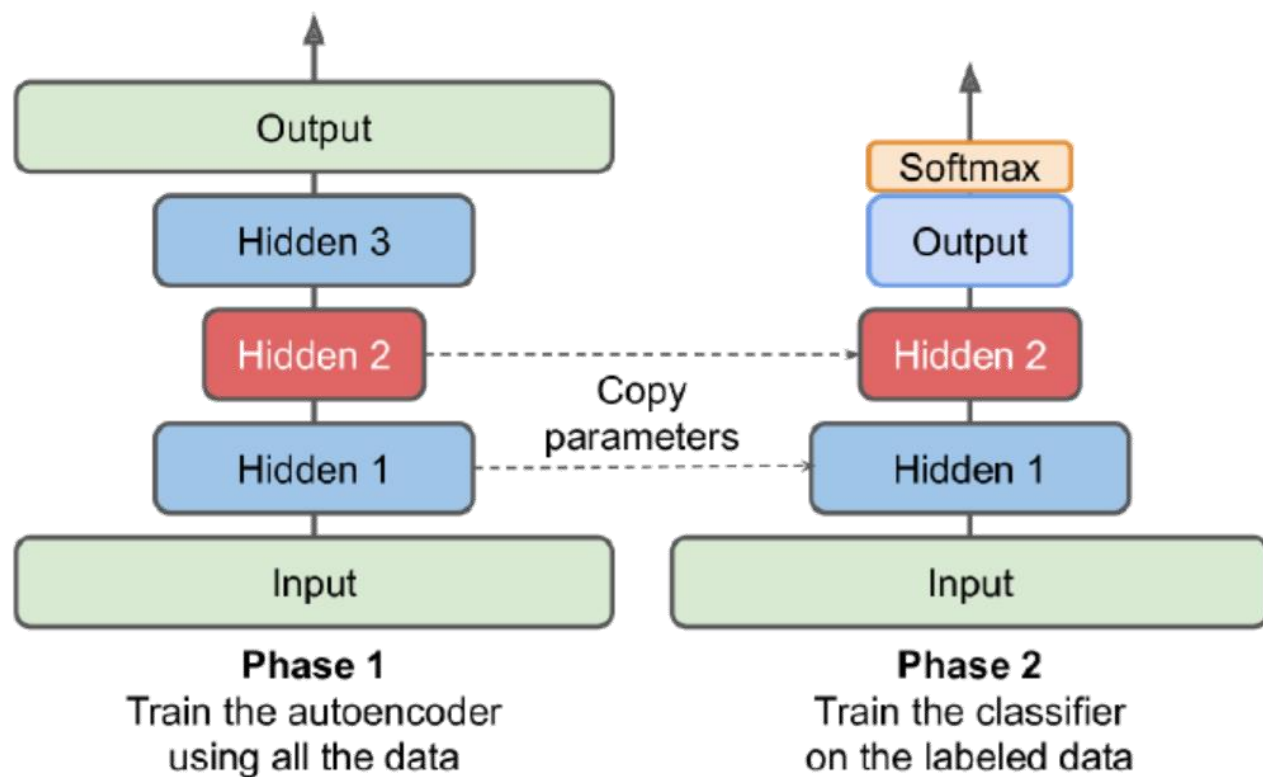


یادگیری بازنمایی

Representation Learning

پیش‌آموزش حریصانه لایه‌ها

- این رویکرد قبل از توسعه تکنیک‌های مدرن برای آموزش شبکه‌های بسیار عمیق (ReLU، بهینه‌سازهای بهتر، معماری‌های بهتر، نرمال‌سازی و ...) انجام می‌شد



- یادگیری بدون ناظر برای بهبود عملکرد شبکه‌های عمیق همچنان پر استفاده است
- به خصوص زمانیکه تعداد داده‌های برچسب‌خورده کم است
- در رویکردهای مدرن، استفاده از داده‌های بدون ناظر تنها برای پیش‌آموزش نیست

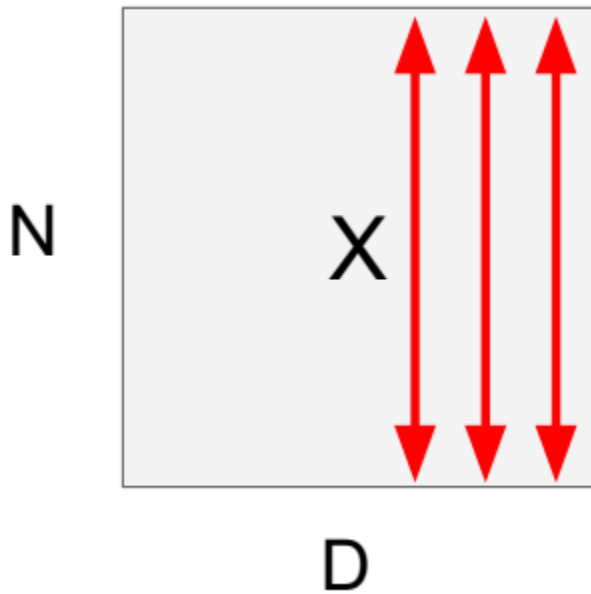
نرمال سازی دسته ای (Batch Normalization)

- نرمال سازی تأثیر چشمگیری بر عملکرد بهینه سازی دارد
- به خصوص برای شبکه های کانولوشنی و شبکه هایی که از تابع غیرخطی سیگموئید استفاده می کنند
- یک دسته از خروجی یک واحد را در نظر بگیرید
 - می خواهیم میانگین آن صفر و واریانس آن یک شود
 - مشتق این تابع به سادگی قابل محاسبه است

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

نرمال سازی دسته ای (Batch Normalization)

- میانگین و واریانس عددی به صورت مستقل برای هر واحد محاسبه می شود
- ورودی: x ($N \times D$)



- میانگین هر کانال (به طول D)

$$\mu_j = \frac{1}{N} \sum_{i=1}^N x_{ij}$$

- واریانس هر کانال (به طول D)

$$\sigma_j^2 = \frac{1}{N} \sum_{i=1}^N (x_{ij} - \mu_j)^2$$

- x نرمال شده ($N \times D$)

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Learnable scale and shift

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

- خروجی: y ($N \times D$)

نرمال سازی دسته‌ای: زمان آزمون

- برآوردهای میانگین و واریانس به minibatch بستگی دارند

- نمی‌توان این کار را در زمان آزمون انجام داد!

- از میانگین متحرک مقادیر (μ و σ^2) در حین آموزش استفاده می‌شود

- در زمان آزمون BN به یک عملگر خطی تبدیل می‌شود!

- می‌تواند با لایه کاملاً متصل یا کانولوشنی قبل ترکیب شود

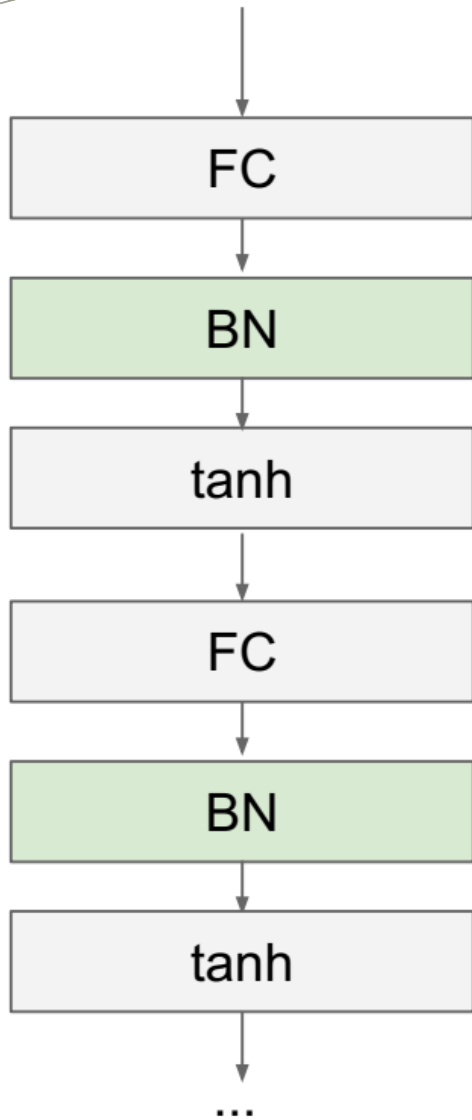
$$\mu_j = \frac{1}{N} \sum_{i=1}^N x_{ij}$$

$$\sigma_j^2 = \frac{1}{N} \sum_{i=1}^N (x_{ij} - \mu_j)^2$$

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

نرمال سازی دسته‌ای



- معمولاً بعد از لایه‌های خطی و قبل از تابع فعال‌سازی غیرخطی استفاده می‌شوند
- جریان گرادیان را بهبود می‌بخشد
- آموزش شبکه‌های عمیق را ساده‌تر می‌کند!
- اجازه می‌دهد از نرخ یادگیری بالاتر استفاده کنیم و همگرایی را سرعت می‌دهد
- حساسیت به مقداردهی اولیه کاهش می‌یابد
- در زمان آموزش به نوعی عمل منظم‌سازی را انجام می‌دهد
- در زمان آزمون سرباری اضافه نمی‌کند
- می‌تواند با لایه خطی قبل ترکیب شود

BN برای لایه‌های کانولوشنی

Batch Normalization for
fully-connected layers

$$x: N \times D$$



$$\mu, \sigma: 1 \times D$$

$$\gamma, \beta: 1 \times D$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

Batch Normalization for
convolutional layers

$$x: N \times W \times H \times C$$



$$\mu, \sigma: 1 \times 1 \times 1 \times C$$

$$\gamma, \beta: 1 \times 1 \times 1 \times C$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

نرمال سازی لایه‌ای (Layer Normalization)

Batch Normalization for
fully-connected layers

$$x: N \times D$$



$$\mu, \sigma: 1 \times D$$

$$\gamma, \beta: 1 \times D$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

Layer Normalization for
fully-connected layers

$$x: N \times D$$



$$\mu, \sigma: N \times 1$$

$$\gamma, \beta: 1 \times D$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

نرمال سازی نمونه‌ای (Instance Normalization)

Batch Normalization for
convolutional layers

$$x: N \times W \times H \times C$$



$$\mu, \sigma: 1 \times 1 \times 1 \times C$$

$$\gamma, \beta: 1 \times 1 \times 1 \times C$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

Instance Normalization for
convolutional layers

$$x: N \times W \times H \times C$$

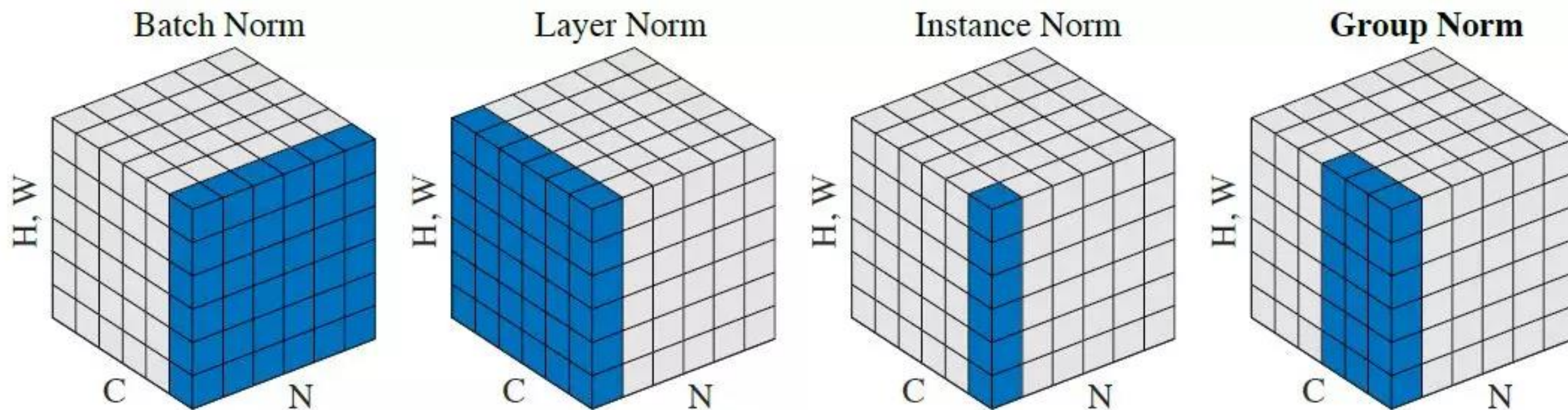


$$\mu, \sigma: N \times 1 \times 1 \times C$$

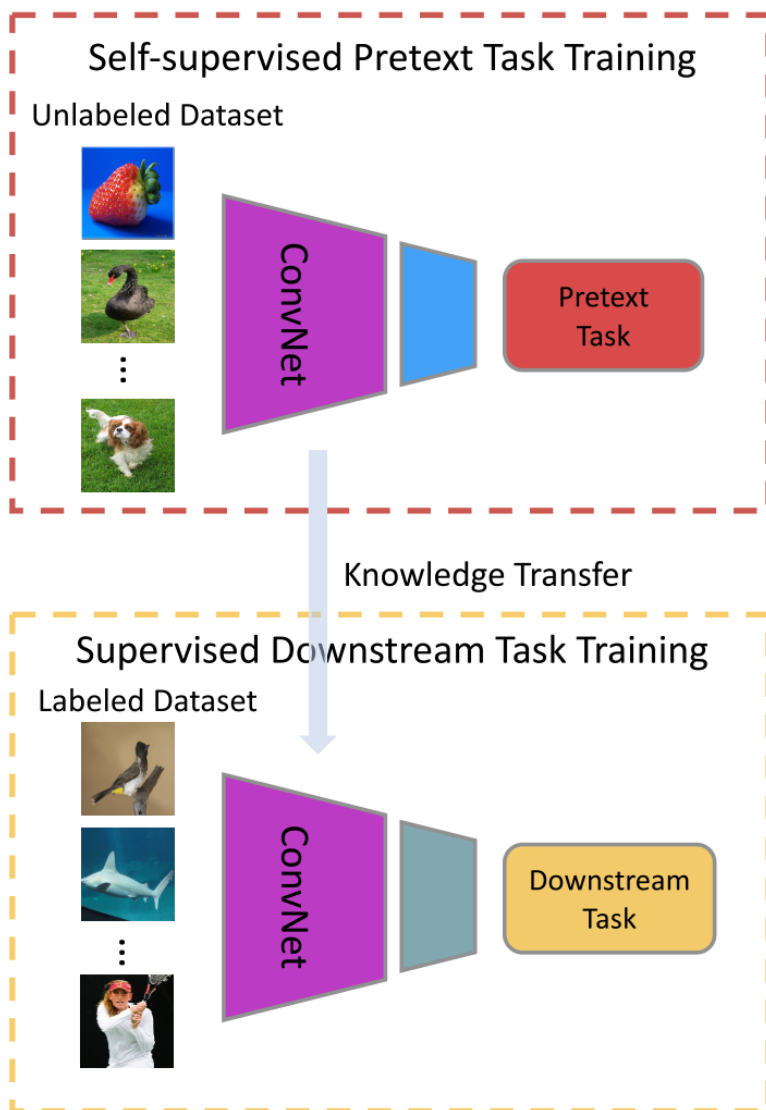
$$\gamma, \beta: 1 \times 1 \times 1 \times C$$

$$y = \gamma \frac{x - \mu}{\sigma} + \beta$$

مقایسه روش‌های نرمال‌سازی



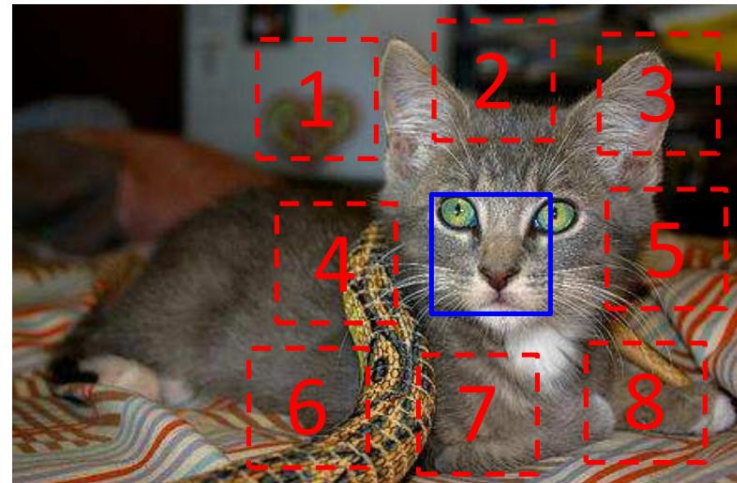
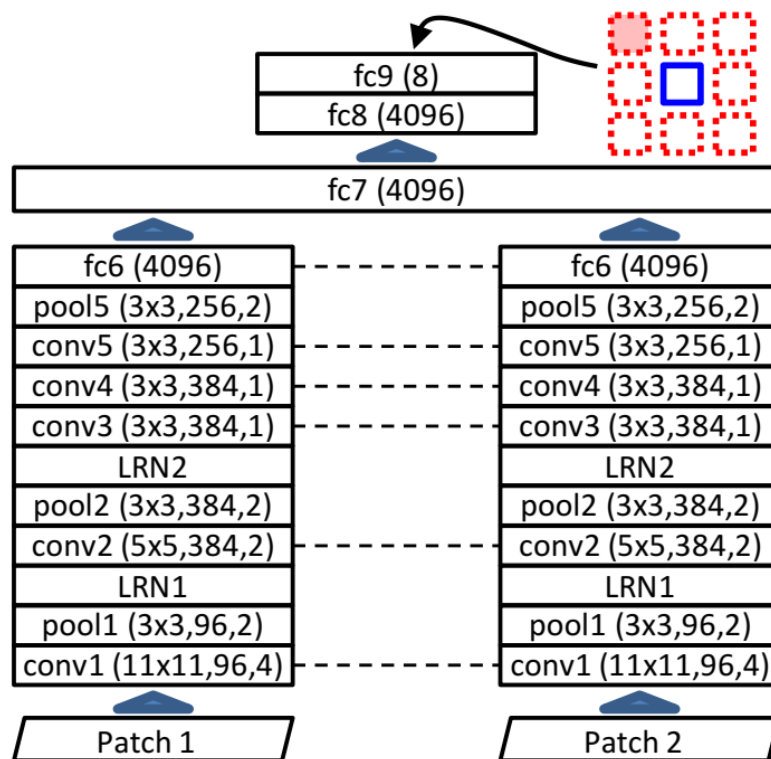
یادگیری خودنظارتی (Self-Supervised)



- روش‌های یادگیری خود نظارتی برای یادگیری ویژگی‌های عمومی از داده‌های بدون برچسب در مقیاس بزرگ پیشنهاد شده‌اند
- مسئله‌های کمکی دو ویژگی مشترک دارند:
 - برای حل مسئله کمکی لازم است تا ویژگی‌های بصری توسط شبکه‌های کانولوشنی استخراج شوند
 - شبه‌برچسب‌های مورد نیاز در مسئله کمکی باید به صورت خودکار برای داده‌های مورد نظر قابل تولید باشند
- این رویکرد برای مسئله‌های غیر از بینایی کامپیوتر مانند پردازش صوت و پردازش متن نیز قابل استفاده است

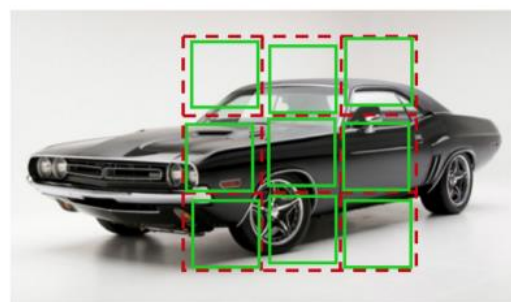
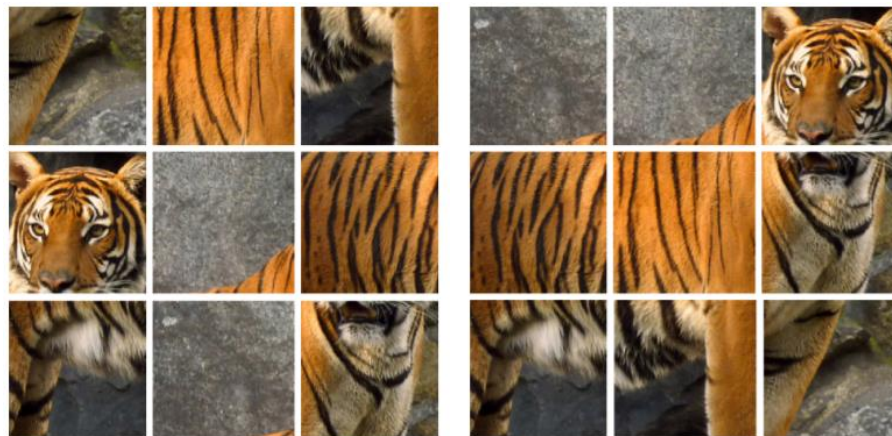
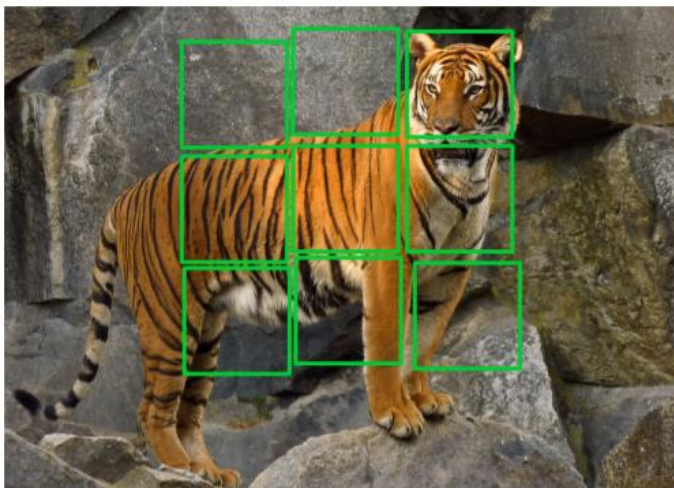
مثال: پیش‌بینی موقعیت

- یادگیری موقعیت نسبی اجزاء تصویر می‌تواند منجر به شناخت خوبی از محتوای تصویر شود
- برای حل این مسئله، نیاز است تا شبکه بتواند اشیاء و اجزاء آنها را یاد بگیرد

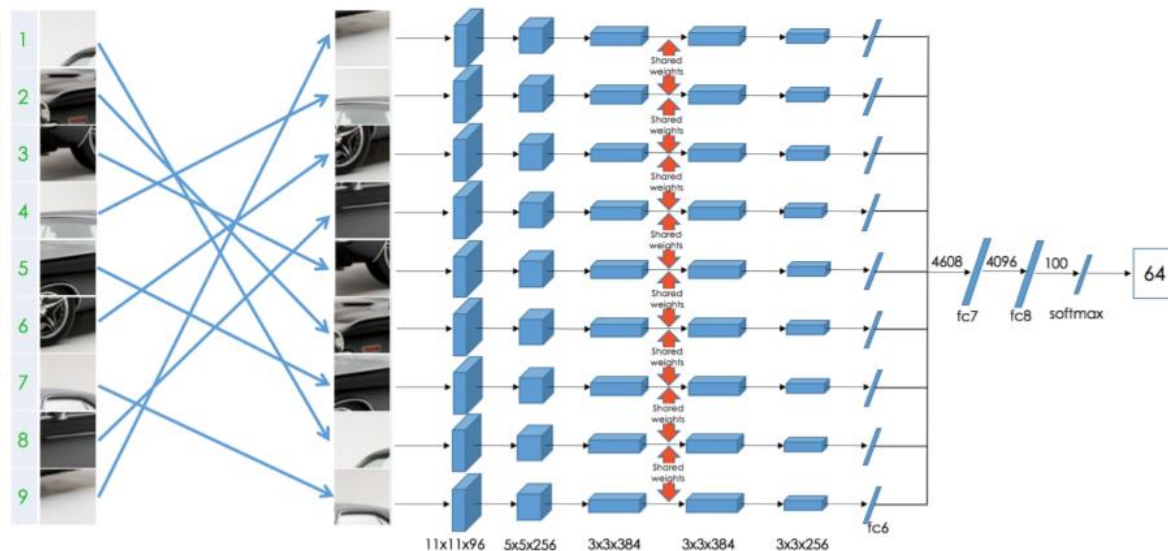


$$X = \left(\begin{array}{c} \text{cat face} \\ \text{cat ear} \end{array} \right); Y = 3$$

مثال: حل جورچین

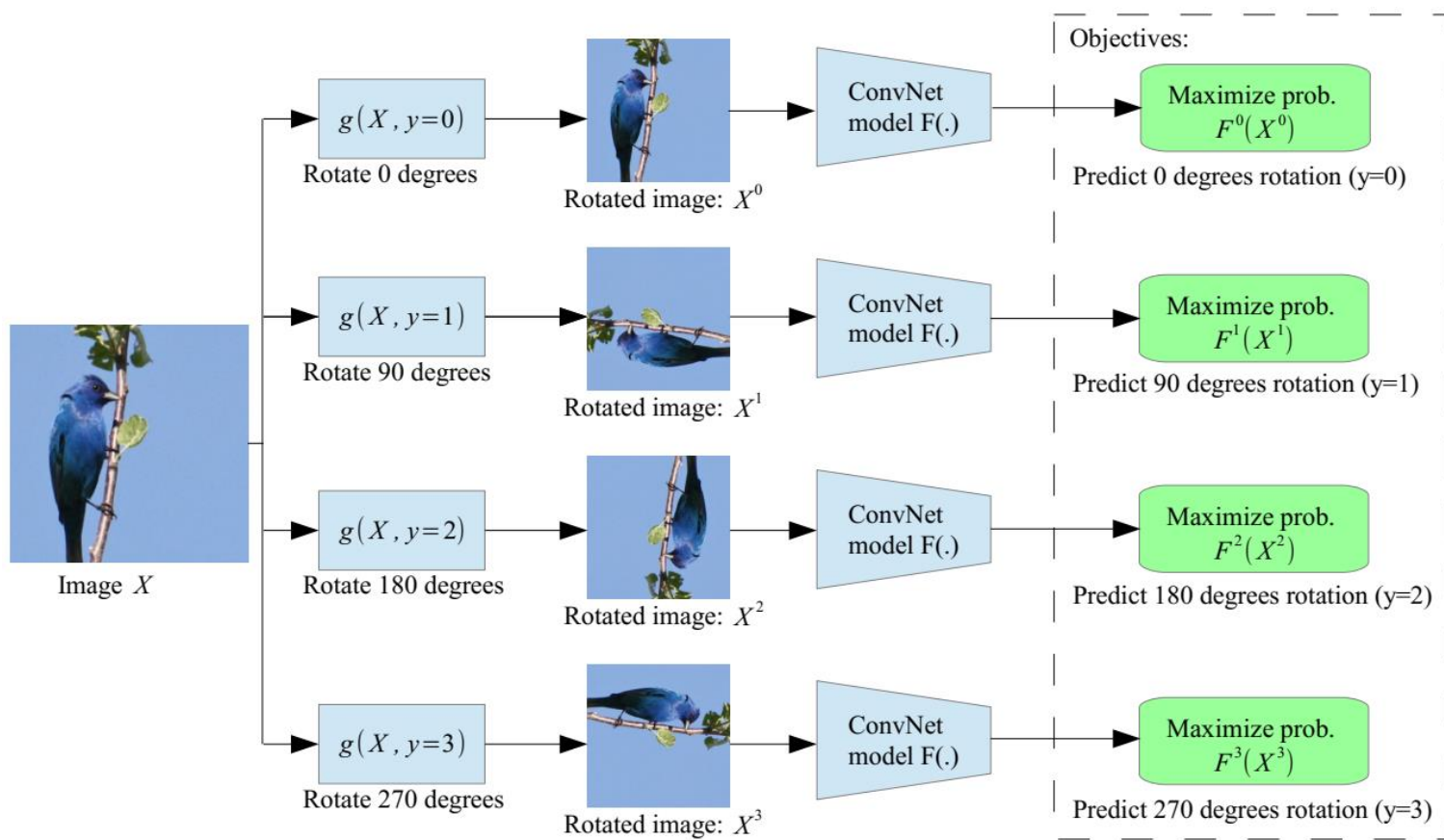


Permutation Set		
index	permutation	Reorder patches according to the selected permutation
64	9,4,6,8,3,2,5,1,7	



مثال: تخمین چرخش

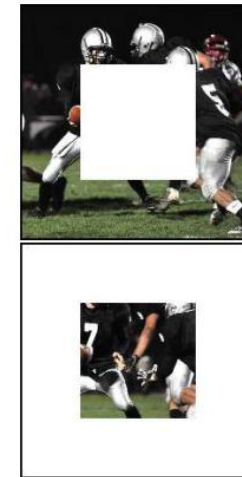
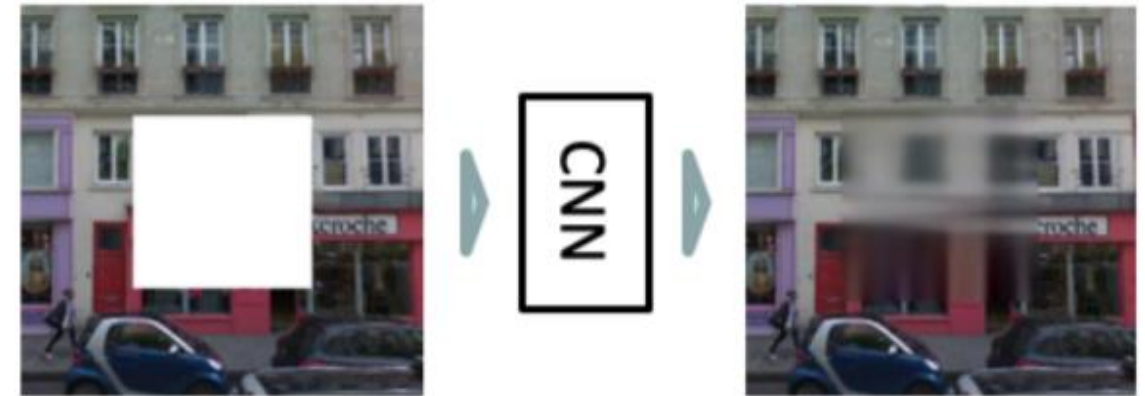
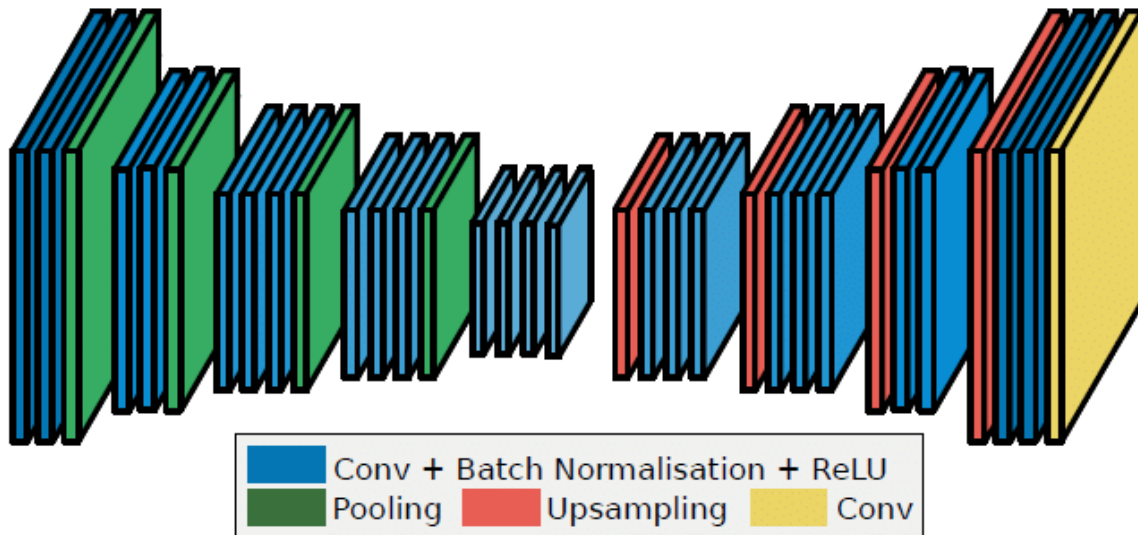
- با آموزش مدل برای تشخیص میزان چرخش ۲ بعدی، ویژگی‌های تصویر آموخته می‌شود



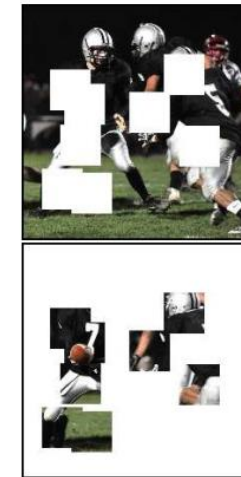
مثال: پیش بینی محتوا

$$\mathcal{L}_{rec}(x) = \left\| \hat{M} \odot \left(x - F \left((1 - \hat{M}) \odot x \right) \right) \right\|_2$$

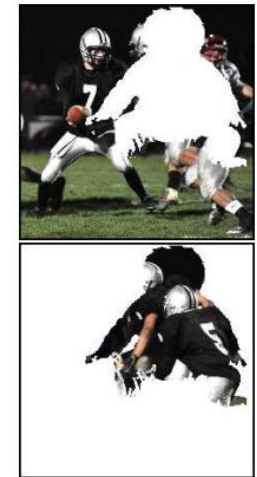
Convolutional Encoder-Decoder



(a) Central region



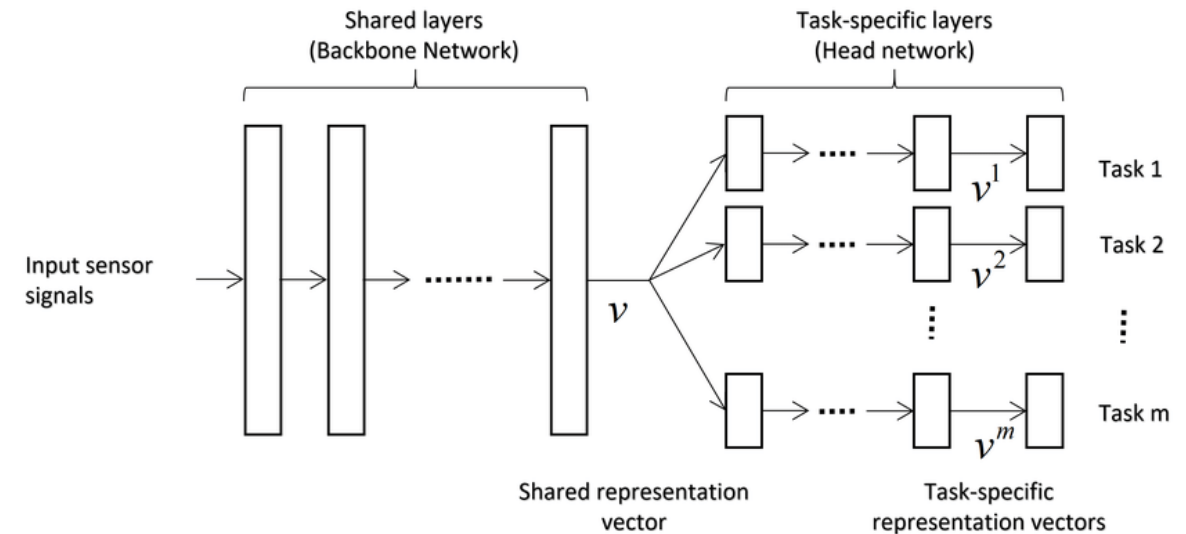
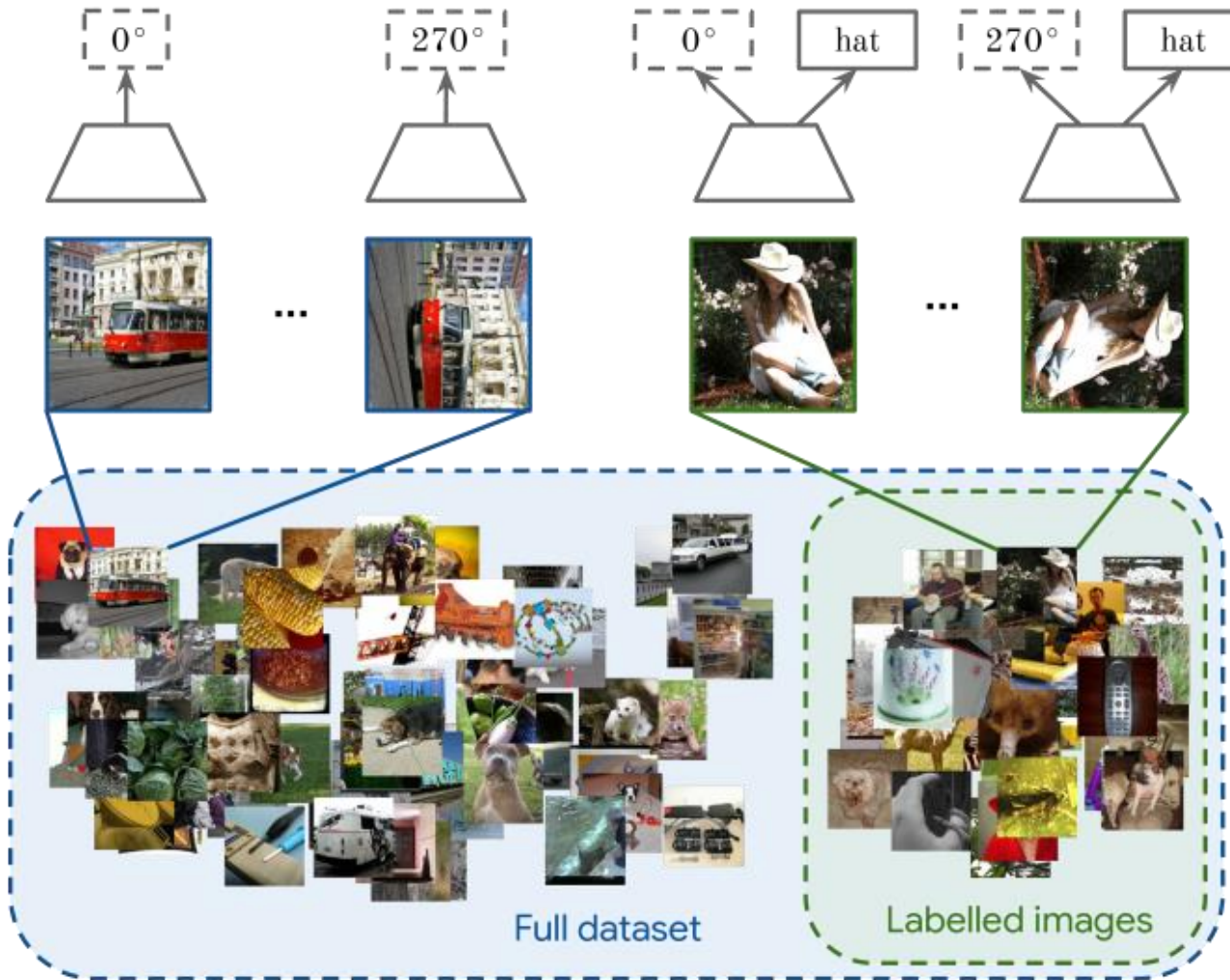
(b) Random block



(c) Random region

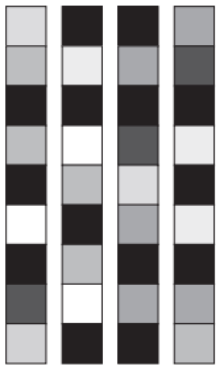
یادگیری چندوظیفه (Multi-Task Learning)

- نتایج نشان می‌دهد که ترکیب وظایف (حتی از طریق یک معماری ساده چند سر) عملکرد را بهبود می‌بخشد



جانمایی کلمات (Word Embedding)

- جانمایی کلمات اطلاعات بیشتر را در ابعاد بسیار کمتری قرار می‌دهد
- این بردارها را می‌توان با استفاده از حجم زیادی از متن پیش‌آموزش داد و در مجموعه داده‌های کوچک از آنها استفاده کرد



One-hot word vectors:

- Sparse
- High-dimensional
- Hardcoded

Word embeddings:

- Dense
- Lower-dimensional
- Learned from data

مدل زبان طبیعی

- در این مدل می‌خواهیم کلمه بعدی را پیش‌بینی کنیم

- هدف ما دستیابی به بردارهای
جانمایی است و وزن‌های دیگر شبکه
هدف نیستند

