Lecture 1

Discriminative modeling estimates p(y | X) Discriminative modeling aims to model the probability of a label y given some observation X / A generative model must include a random component that influences the individual samples generated by the model.

Diagram

Description automatically generated

Generate completely novel observations ● New feature combinations should mimic training data patterns ● Enormously challenging due to exponential combination possibilities ● Vast majority of arrangements don't resemble plausible observations ● Model must have randomized component to produce variation ● Cannot be fixed calculation like averaging feature values

**Accuracy** If pmodel(x) is high for a generated observation, it should look like it has been drawn from pdata(x) If pmodel(x) is low for a generated observation, it should NOT look like it has been drawn from pdata(x) **Generation** It should be possible to easily sample a new observation from pmodel(x) **Representation** It should be possible to understand how different high-level features in the data are represented by pmodel(x)

representation learning tries to find the lower-dimensional latent subspace or manifold on which particular kinds of image lie (for example, the dog manifold)

A probability density function is a function that describes the relative likelihood for a continuous random variable to take on a given value. The integral of the density function over all points in the sample space must equal 1, so that it is a well-defined probability distribution / A parametric model is a family of density functions ptheta(x) that can be described using a finite number of parameters theta

Generative modeling can be thought of as a form of maximum likelihood estimation. Approaches: Explicitly model the density function, but constrain the model in some way, so that the density function is tractable (i.e., it can be calculated). 2. Explicitly model a tractable approximation of the density function. 3. Implicitly model the density function, through a stochastic process that directly generates data. Implicit density models do not aim to estimate the probability density at all, but instead focus solely on producing a stochastic process that directly generates data. ● The best-known example of an implicit generative model is a generative adversarial network. ● We can further split explicit density models into those that directly optimize the density function (tractable models) and those that only optimize an approximation of it. / Tractable models place constraints on the model architecture, so that the density function has a form that makes it easy to calculate. / Approximate density models include variational autoencoders, which introduce a latent variable and optimize an approximation of the joint density function. / Diffusion models approximate the density function by training a model to gradually denoise a given image that has been previously corrupted

Lecture 2

A vanilla neural network: Flatten, Dense(200) (ReLU), Dense (150) (ReLU), Dense (10) (Softmax) – generates the likelihood of a picture.

CNN Convolutional layers were designed specifically for images. ● They operate in two dimensions and can capture shape information; they work by sliding a small window, called a convolutional filter, across the image in both directions. ● A convolution is an mathematical operation between a small matrix (kernel) and a larger matrix (input) ● The kernel is slid across the input matrix ● At each spatial location, the dot product is computed between the kernel and a patch of the input ● This dot product result forms the output activation map ● Convolutions enable efficient localization and extraction of patterns / A single convolutional filter can process an entire image with very few learnable parameters ● It will not be able to learn and represent enough of the complexities of the image and multiple filters are needed. ● A convolutional layer typically contains tens or hundreds of similar filters, each with its own independent learnable weights ● They are applied to the image in succession, and each produces a channel of output values. ● The output of a convolutional layer is a multichannel set of 2D values.

Learnable weights = filter size (e.g., 4x4) X Input Channels X Number of Filters / Note that Number of Filters = output channels / downsampling, commonly a 2x2 downsampled by max or average / Flatten layer has no parameters / AlexNet uses 3x3 max-pooling operations with a stride of 2. / When the ReLU activation function is used, it is customary to initialize the bias to a small positive value before training so that, after activation, all layers start with a nonzero output and a nonzero gradient / with many layers, the gradients tend to be spread too thin across all layers and the network converges slowly or not at all.

Skip connections combine the input (X) with the convolution of x (C(X)) to get f(X) = C(X) + X / Skip connections seem to help gradients flow through the network during the optimization (backpropagation) phase.

AlexNet, ResNet, DenseNet, MobileNetv2 – millions of parameters. Transfer Learning. Typically only the fully connected/output layer is changed and retrained / The weights of the pretrained layers are frozen as they contain more general features useful for the new task. Fine Tuning Doing full fine-tuning of all layers requires more data and compute than transfer learning, but can adapt the pretrained features better to the new task and achieve higher performance. Challenges: ● Learning rates too high: Destroy pretrained knowledge ● Learning rates too low: Very slow training convergence / Low LR for early layers, higher for later layers

SimCLR, or Simple Framework for Contrastive Learning of Visual Representations, is a groundbreaking approach to self-supervised learning in the domain of computer vision. The most effective combination found was random cropping followed by random color distortion.

SimCLR introduces the NT-Xent loss function, which is a temperature-scaled cross-entropy loss that operates on the cosine similarity between representations.

Diagram

Description automatically generated with medium confidence

Lecture 3

Combination of gaussians. Each has a mean and standard deviation. Each is weighted. Probability of x is given by the sum of all gaussians for all values of n. The sum from n=1 to N of lambda n times Norm[mean n , sigma n ]. Probability of x given that latent variable z takes on value n. Infinite sum of spherical gaussians. Ancestorial sampling. Concentrate on leaning theta. Assume SD is known. To train the model, we maximize the log-likelihood. Need ELBO. ELBO is always less than the log-liklihood

Chart

Description automatically generated with medium confidence

KL divergence measures difference of two probability distributions, always greater than 0, at best identical

Adding some "fuzziness" to embeddings while regularizing helps autoencoders achieve better creative generation.

Graphical user interface, diagram

Description automatically generated

True posterior p(z) is complex and intractable ○ Variational inference uses a simpler approximate posterior q(z) ○ Optimizes q(z) to be as close as possible to p(z), Variational autoencoders assume that there is no correlation between dimensions in the latent space. The decoder of a variational autoencoder is identical to the decoder of a plain autoencoder / In an autoencoder, each image is mapped directly to one point in the latent space. ● In a variational autoencoder, each image is instead mapped to a multivariate normal distribution around a point in the latent space. Reparameterization trick: Rather than sample directly from a normal distribution with parameters z\_mean and z\_log\_var, we can sample epsilon from a standard normal and then manually adjust the sample to have the correct mean and variance.

Plain autoencoders map images to a latent feature space ● But sampling the learned space struggles to generate realistic novel images ● Variational autoencoders solve this by: ○ Introducing randomness ○ Constraining the latent distribution ● Transforms the autoencoder into a powerful generative model ● Decoding random points synthesizes new examples ● The structured latent space also enables: ○ Intuitive feature vector arithmetic ○ Face morphing and editing applications

Lecture 4

Query, Key, Value – Attention(Q,K,V) = softmax(QKt / SQRT(dk))V

The two most commonly used attention functions are additive attention, and dot-product (multiplicative) attention. ● Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. ● While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code. ● While for small values of dk the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of dk . ● For large values of dk , the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients .

Inputs are encoded and positionally encoded. PEeven = sin(pos/100002i/dmodel), PEodd = cos(pos/100002i/dmodel)

Mult head results are concatenated and then pass through W0 to reduce back to correct size. Layer normalization on input and each layer, outputs have mean 0 and SD 1, enables faster convergence, add epsilon to sigma to not get divide by zero

Encoder only – BERT, RoBERTa and DistilBERT, pre taining and fine tuning, MLM – masked language model – randomly mask tokens

Decoder Only – T-DMCA - The multi-head self-attention is modified to reduce memory usage by limiting the dot products between Q and K. Local attention: Sequence tokens are divided into blocks of similar length and attention is performed in each block independently. ● As the attention memory cost per block becomes constant, this modification allow us to keep the number of activations linear with respect to the sequence length. ● Memory-compressed attention: Reduce the number of keys and values using a strided convolution. ○ The number of queries remains unchanged. ○ This modification allows the Transformer to divide the number of activations. ○ Convolution kernels are of size 3 with stride 3. ○ The local attention layers capture the local information within a block. However, the memory compressed attention layers are able to exchange information globally on the entire sequence. Encoder-decoder – BART, T5 / mask out future words by setting attention scores to – negative infinity

What is the purpose of an autoencoder? Generate new images, Compress and reconstruct data / In a VAE, the reconstruction loss measures the difference between the input and reconstructed image / Implicit density models focus on directly estimating the probability density function FALSE / What is the main aim of discriminative modeling? To predict labels or classes for given data points