Cours

Charles Vin

Date

1 Feature detection & description

- Local detection/description, looking for invariance
- feature detection : Points/Regions of Interest detection : Corner detection
  - Detection of change in two directions with the eigenvalue of the Hessian matrix (one value = one direction)
  - Convolution with special filter $\rightarrow$ Large value = corner + comparaison with value before moving the windows $\rightarrow$ Threashold $\rightarrow$ corners
- Feature description :
  - Example : SIFT
  - Similar patch of image == close descriptor
- Bag of visual word of descriptor / features
  - Extraction (with feature descriptor or cnn) $\rightarrow$ into clustering (unsupervised learning ; K-Means, GMM, ...)
  -
- K-Means
  - Loss : $C(w) = \sum_{i=1}^{n} \min_k \|x_i - w_k\|^2$
  - pros : simplicty, convergence to local min
  - Cons : mémory intesive, choice of K, sensitive to init and artefacts, pherical clusters
- Image signature (not sure about this one) : matrix of Likelihood value, size $M \times K$ with K dico size, number of cluster extracted and $M =$ number of feature, then take the maximum likehood I think

2 SVM

- Problem : Donnée non linéaire $\rightarrow$ Projection, mais si projection dim ++ $\rightarrow$ Attention sur apprentissage + quel dim choisir $\rightarrow$ Solution : SMV do this auto
- Maximiser la marge $\gamma \Leftrightarrow$ minimiser $\|w\|$ sous la contrainte $\forall i, (wx^i + b)y^i \geq 1$ par des calculs obscures (≥ 1 car on veut que la distance entre la droite de régression et ces deux marges soit supérieur 1)
- Prise en compte des erreurs si pas de frontière linéaire pur : (soft margin)
  - ξ variable de débordement par rapport à sa marge pour chaque point mal classé $\rightarrow$ Raison obscure $\rightarrow \xi = \max(0, 1 - (wx^i + b)y^i)$ Hinge loss
  - On avait $\min ||w||^2$ maintenant $\min ||w|||^2 + K \sum \xi$ avec K hyper param nombre d'erreur
  - $\|w\|^2$ = margin maximization, $K \sum \xi$ = Constraint satisfaction
  - $\|w\|^2$ = Régularization, $K \sum \xi$ = Data fitting
- The support vectors are the data points that lie on the margin, which is the region between the decision boundary and the closest data points of each class. Support vectors are critical in SVM because they determine the location and orientation of the decision boundary. All other data points that are not support vectors are not used to construct the decision boundary, which means that SVM is robust to noise and outliers in the data.
- La taille de la marge est un hyper-paramètre important : marge grande == underfitting // marge petite == overfitting (séparation linéaire plus proche des points, moins centrée)
- K petit = $K \sum \xi$ petit = petite pénalisation des erreur = tolérance de celle ci = underfitting // inverse
- Better on noisy problem

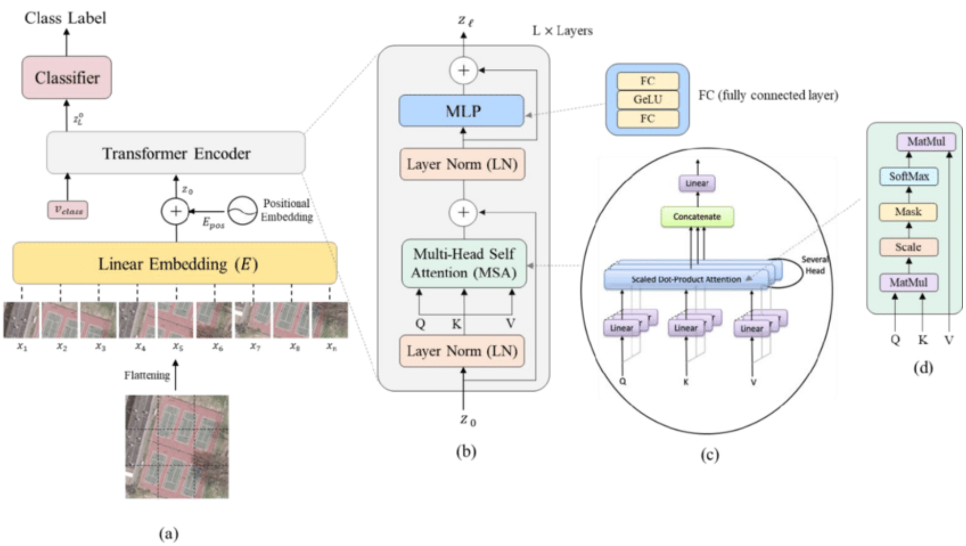
Kernel Tricks :

— Kernel Function : $k(x, y) =< \phi(x), \phi(y) >$
— Mesure la similarité entre 2 objets
  — – = vecteur opposé = éloigné
  — = 0 = produit orthogonal = éloigné
  — ++ = vecteur aligné = proche
— Stable pas addition, multiplication, composition avec f polynome, exponentiel
— La complexité de calcul d'un noyau polynomial est linéaire par rapport d le degré du polynome. Mais pas la dimensionnalité de la projection
—

3 CNN arch

— LeNet5 : first CNN, for MNIST, FC at the end
— AlexNet : bigger, GPU, more data, dropout, pooling, relu, data aug, contrast normalization, FC at the end
— GoogLeNet : Inception layers, auxilary classifier + main classifier, less FC than alexnet at the end of each head (less params)
— VGG : rly deep, the deeper the better
— ResNet : good of deep, skip (residual) connection to stabilise training

4 ViT



— ConvNet = local attention (less local after many layers) VS ViT = global attention

5 GANs

— Auto encoder & VAE
  — problem : one pixel difference beweet generated image and the target can be either realistic or not
— Discriminator : generated image VS real images :
— Generator : random image → generation → discriminator
— Learning :
  — Discriminator : GD with a freeze generator
  — Generator : GD with a freeze discriminator
  — loop 50.000 times

- Math and Loss

$$V(G, D) = \mathbb{E}_{x \sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{x \sim P_G}[\log(1 - D(x))]$$
$$G^* = \arg \min_G \max_D V(G, D)$$

  - For the generator $\max_D V(G, D)$ evaluate the "difference" between P_G and P_{data}
- Evaluation :
- Cons : Learning can be hard : G and D must be well synchronized for convergence
- Pros : Computationnal efficient (no complexe likelihood inference), can fit sharper distribution, Spatial resolution, object quality
- Achitecture improvement
  - Laplacian Pyramid GAN (LAPGANs) (improve spacial resolution) : improve the generator : combines the strengths of Laplacian pyramids and GANs. It generates images in a coarse-to-fine fashion, where each level of the Laplacian pyramid refines the image details, leading to high-quality, high-resolution image generation.
  - DCGANs (Improve object quality) : full conv generator and discriminator (no fcc, better activation fnc, batchnorm); upsampling step by step
  - ProGANs : combine both idea : the network is trained incrementally, starting with low-resolution images and progressively increasing the resolution by adding layers to the network. This approach enhances the stability and efficiency of the training process, allowing the generation of high-resolution, detailed images with improved quality and variation. $\rightarrow$ complicated to train "out of the box" (block adding logistics and have to ajust params for each dataset)
  - MSG-GANs : upsampling + skip connection between the generator and the discriminator for better learning
  - StyleGANs : style transfer thing with control of the image generation process at different levels of detail through the use of adaptive instance normalization (AdaIN)
- Editing : Possible to change a generated image in the latent space to move it in a different class zone with linear interpolation (smiling womand - neutral women + neutral man = smiling man)