1. Artificial Intelligence → strong AI / Artificial General Intelligence (AGI)

Machine learning (ML)
- Deep learning (DL)
- Reinforcement learning (RL)
- △ Predictive learning (PL)

- ANN (Artificial Neural Network)
- BNN (Biological Neural Network)

应用：
- speech recognition
- computer vision
- Natural language Processing

架构：CNN. RNN → LSTM / GRU (Gated Recurrent Unit)

*. AI的泛化能力 (Generalization ability). 应对拟合能 (void over fitting). ← 拟人脑与自然界的逐层抽象结构. 训练与推断的速度问题 (speed of traning and infering)

*. AI从传统的多阶段分治法转向更简单便捷的端到端训练 (e2e).

*. DNN的特点：
① 数据与算力
② 从特征工程到逐层抽象
③ 统计不必要. △. 人类更擅长逻辑和规律

*. 人工智能流派
① Symbolism. 符号主义. 即计算主义 / 逻辑主义. eg. 专家图谱
② 统计主义 → 连接主义. eg. CNN.

2. 卷积 DCNN.

MLP. 表现不佳 (capacity 太大的出现). 亦叫 fully connected NN.
① 参数多. 容量大. 拟合能力强. 即大 capacity.
② 并行计算特性
③ 可训练性：BP (backpropagation). SGD (stochastic gradient descent)
*. 多参数小样本下防止过拟合的能力.
*. batch_size · num_batch = dataset_size
△. batch_size. mini-batch. epoch

No.

Date

<1>. 深度学习框架

computation graph. { symbolic. 静态图
imperative. 动态图

eg: CUDA + cudnn + MXNet + GraphViz

*. BP算法基本原理. 梯度计算

*. 梯度消失与爆炸

Batch-Normalization. ResNet. gradient clipping (梯度截断)

gradient penalty (梯度惩罚)

△. 非线性激活函数的重要性

*. 更宽. 更深的网络不能忽略

*. 调参的重要性

△. 拓扑学. 任何n维流形均可在 2n+2 维空间中划分开

<2>. 一些理论和概念

① dataset { training data. 找参数
validation data. 找超参数
test data. 评估泛化能力

*. cross validation

② { supervised learning { regression → MSE
classification → cross-entropy
tagging (multi-classes classification)
seq2seq { translation
speech recognition

eg. RNN. LSTM.

unsupervised learning { clustering.
dimensionality reduction
generative model: auto-encoding. GAN
recommendation { 协同过滤
关联规则

semi-supervised learning. 多用于数据标注工具

③ { under-fitting. capacity不足. 节点设置不合理. 样本数据问题
over-fitting { regularization
data augmentation

*. early stopping.

④. sgd. → 贪心算法 (greedy algorithm).

面对 non-convex 优化问题 { 陷入 local minimum / 无法找到 global minimum } under-fitting.

*. saddle point (鞍点). → 训练停滞.

解决方法. momentum. Adagrad. Adam. etc.

⑤ 正则化 (Regularization) 方法解决过拟合问题.

A. L1. L2 regularization

△. 奥卡姆剃刀原则: 愈简单愈好.

$$\begin{cases} L1_loss = Loss + \lambda \sum_w |w| \\ L2_loss = Loss + \lambda \sum_w w^2 \end{cases}$$

eg. 取 L2 正则化时.

由 $w_{new} = w - \eta \cdot \frac{\partial Loss}{\partial w}$ 有: $w_{new} = w - \eta \cdot \frac{\partial Loss_L2}{\partial w}$

$= (1-2\eta\lambda) w - \eta \cdot \frac{\partial Loss}{\partial w}$

即其中正则项入的作用即让 w 在训练中每一步都乘因子 $(1-2\eta\lambda)$. 从而略衰减. 也而趋到 w 趋向于 0.

B. dropout 方法.

利用那些因数的神经元充分发挥作用. 得到训练.

C. ensemble. 多模型集成.

D. 多任务学习.

E. 参数共享. △. CNN 的主要特征之一. 此外还有 BN 层. batch-normalization

F. early stop

G. label smothing (标签平滑) 与 noise label (噪声 label).

{ 原始样本: $(x_i, y_i)$. $(x_j, y_j)$

虚拟样本: $(\lambda x_i + (1-\lambda) y_j, \lambda y_i + (1-\lambda) y_j)$ }

⑥ { data preprocessing { 归一化 (normalization) / 白化 (whitening): PCA. ZCA. / 直方图均衡 (histogram equalization) }

data augmentation. 镜像. 平移. 旋转. 放缩. 剪切. 裁剪. 仿射变换和仿射组合. 色彩调整. 加入噪声. 图形区域擦除.

⑦ 调参: grid search / random research.

△. learning_rate 衰减 / batch_size → 成比例增长.

△. 均值为0. 方差$\sigma^2$调整.

初始化方法：随机初始化（正态分布参数）

△. pretrained model： pre-training + fine-tuning.

*. https://playground.tensorflow.org.

<3>. MINIST实例

△. softmax. 归一化概率.

$$\text{softmax}(x_i) \rightarrow y_i = e^{x_i} / \sum_{i=1}^{n} e^{x_i}$$

使得输出值在0到1之间且和为1.

△. sigmod. $\text{sigmod}(x_i) \in (0,1)$.

△. 交叉熵损失函数：

$$loss_{CE} = -\sum_i p_i \log q_i$$，其中$p_i$为label. $q_i$为预测概率.

<4>常见bug与调试方法.

data bug / model bug / training process bug.

3. CNN卷积网络

CNN优势：
- 更符合图像特征的本质
- 自我正则化能力
- 参数共享

△. defination of convolution operation.

[convolution（线性操作）+ bias] → 非线性激活.

△. $n' = \frac{n-k}{s} + 1$.（n为原图. k为kernel size, s为步长）

△. 池化
- 最大池化
- 平均池化
- 全局池化

} 缩小图像

△. 转置卷积（反卷积）：放大图像

deconvolution / transposed convolution

*. batch-size 越大. 训练越快. 精度稍低.

△. MXNet使用技巧.

①函数式快速定义多个层. 组成神经网络.

②保存与载入网络结构. 参数.

③多线程异步载入数据. *. eg. MXNet.io.RecordIO / ImageRecordIter.

△④. MXNet中model. 计算的有向图细节. 实现对训练过程的精细控制.

△. 在Web浏览器与移动端运行神经网络的实现.

No.

Date

△. MXNet工具的使用：

amalgamation. 将python模型与代码打包为一个c文件. 然后编译成其它语言进行封装运行.

。deep-learning-model-convertor. 不同框架模型转换.

<1> 经典的深度卷积网络架构

① 深度学习革命的开幕者. AlexNet.

data augmentation. 256×256. 从中心与四角crop 224×224. 镜像.

△. 参数量 = N×5×2 = 10·N { 测试时平均值 / 训练增加参数

Batch-Normalization / dropout. 防止过拟合

△ { convolutional layer 卷积少. 计算量大
fully connected layer 卷积多. 计算量小. → 内存都会消

现代网络多用 convolutional structure. △.

② VGG.

{ 3×3. stride=1. padding=1. → 保持图像大小不变. 方便加深网络
1×1 kernel. 减少计算量.

③ darkNet. 抛弃全连接结构.

△. ④ ResNet与DenseNet中的跳跃连接.

<2> 网络可视化.

<3>. fine-tuning / transfer learning.

△. one/zero-shot learning

目标数据量较少时可以考虑固定网络前n层. 只训练高层的语义表达层.

<4> 架构技巧.

① receptive field.

{ k, k kernel → output中一个像素与input中k×k个像素相关.
△. 这样模拟可通过多个小卷积多层结构获得大卷积核一样的感受野.
△. 同时节省参数量.
同时每层之后加入非线性激活. 进一步提高系统非线性拟合能力.

△. Google提出的Inception结构即是如此

② 1×1 convolutional kernel.

N个1×1卷积核用很少的参数将M张图变为N个图.

1>. 分N类时. 用1×1卷积核将通道数变为N. 再全局池化为N个数字. 再后softmax即可.

2> 用小卷积核层. bottleneck layer：节省参数. 提高系统非线性性能.

eg. { 256 channels → 256×(3.3) kernels → 256 channels
{ 256 channels → 64×(1.1) → 64×(3.3) → 256×(1.1) → 256 channels

网络参数变到11.8%. 表达能力提高

3>. 改变通道数或图像尺寸.

4>. 巧妙使用 1×1 卷积. 即在每像素上实现一个小型MLP网络.

△. 这是 Network In Network 的思想.

⑦. Batch_Normalization. 批规范化.

△. Conv → BN → nonlinear → next Conv

主要作用是使网络中间数据分布规范化. 也即:

1> 加速深度网络训练.
2> 拥有一定的正则化能力. 防止过拟合.
3>. 防止梯度消失、梯度爆炸.

BN: $y = \gamma \cdot \frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}} + \beta$ 即使数据服从正态分布 $N(x|\mu,\sigma^2)$.

其中 $\gamma$ 与 $\beta$ 为可训练参数.

<5> Residual Network.

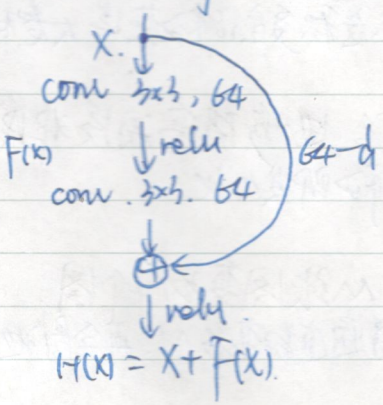
主要思想:

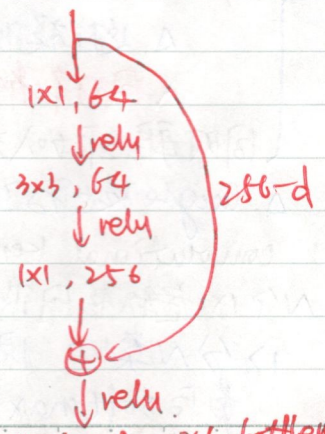
1> 假设网络M. 输出为 X.
2> 给M增加一层变为M'. 则新的输出为 H(X).
3> 有一种方法保持M'的性能至少不会比M差. 即从M与M' 保持恒等变换 (identity transformation). 也即 H(X)=X.

△. 这就保证了网络加深之后性能肯定不会变差.

4> 进一步令 H(X) = X + F(X) (F(X) ≥ 0). 即用新的层去学习 F(X). △. 这里F(X)即为残差 (residue).

△. basic block of residual network.




△. residual block with bottleneck layer

△. 残差结构中Relu层的位置. pre-act 结构. 体现了信息畅通无阻 对梯度流的重要性.

$H(x) = x + Relu(F(x))$

即对原始ResNet结构 $H(x) = Relu(x + F(x))$ 中Relu位置的调整

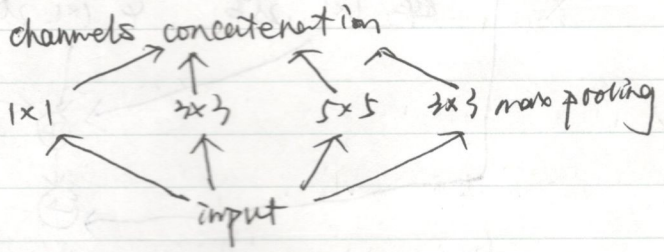
△. 有n个残差单元的网络. 有$2^n$种信息通路. 可理解为$2^n$个浅层网络的集成学习.

△. 随机深度. (Stochastic Depth Method). → 类比 Dropout

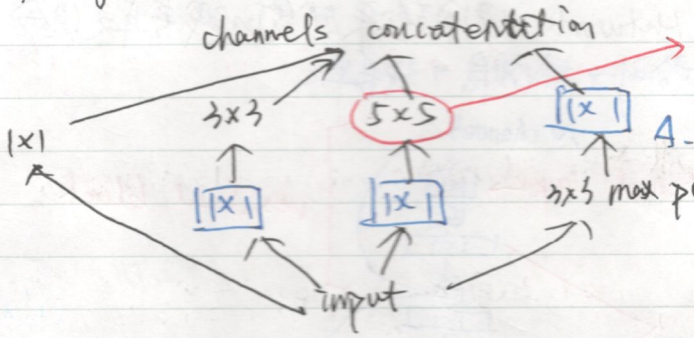
在训练时随机丢弃一些层. 而在inference时全部使用.

<6> 通道组合: Inception block.

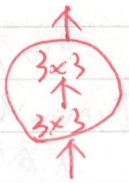
通道组合:




带瓶颈层的改进通道组合:



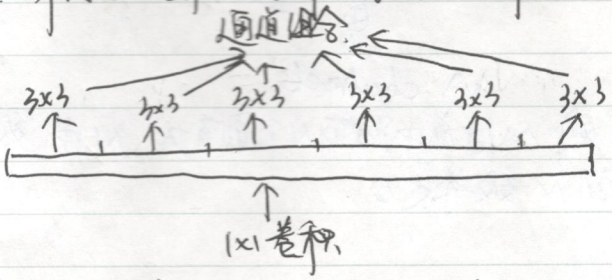

△. 5x5 conv 可以进一步拆分为两个3x3 conv.




△. 1x1卷积为bottleneck layer. 降低通道数以减少计算量.

△. nxn卷积亦可拆分为1xn与nx1卷积.

<7>. Xception. 深度可分卷积. (depthwise separable convolution).




△. Xception Architecture

① Xception结构内部无Relu. 在Xception模块合并加入Relu增加非线性.

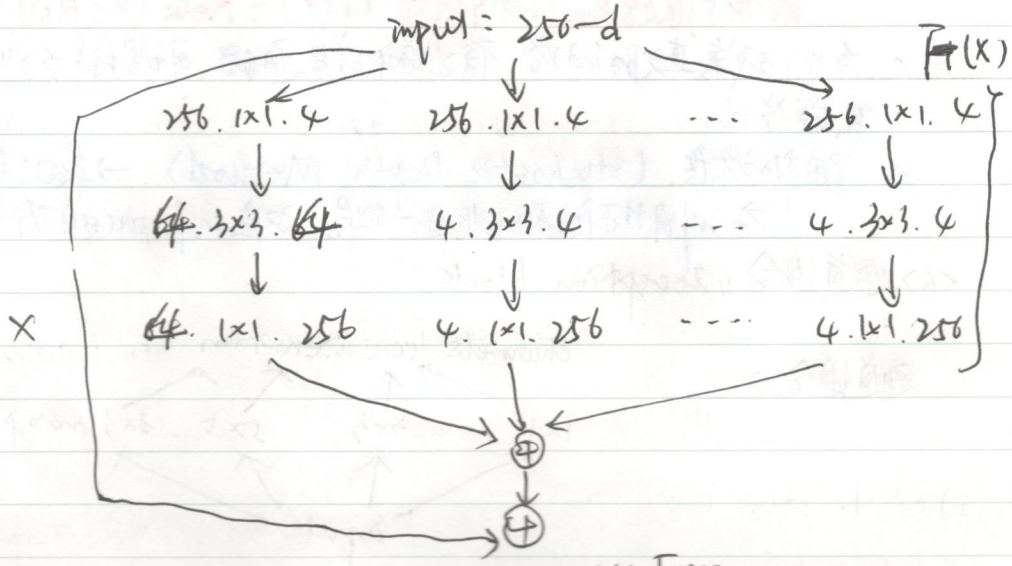
△. 内部的Relu会阻碍信息流动.

② 3x3 kernel 独立运作在每个通道中. 1x1 kernel 负责通道间的组合.

△. 参数量. 仅有10%左右的参数量. 很大程度上获得几乎相同甚至更高的准确

ResNet的演进变种.

<8>①ResNext. 结合ResNet与Inception的思想. 在残差结构中嵌入通道组合模块.

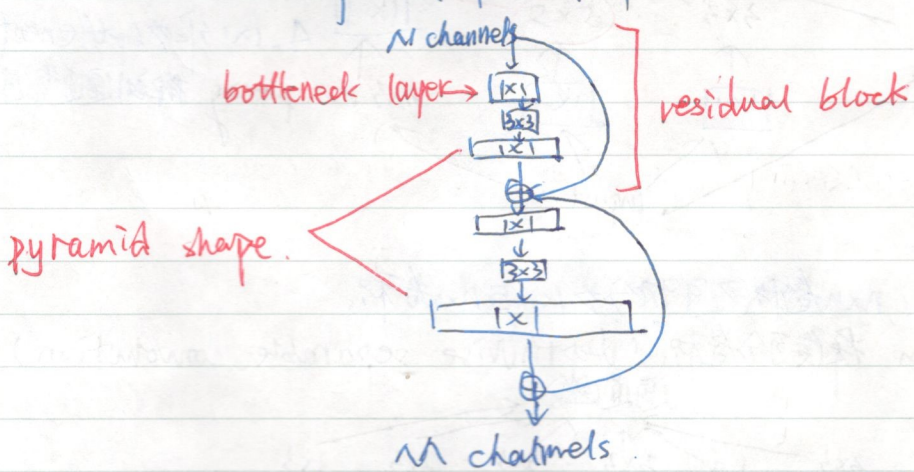



$H(x) = x + F(x)$.

其中 A. NxN. B 表示. 输入A通道. NxN卷积. 输出B通道.

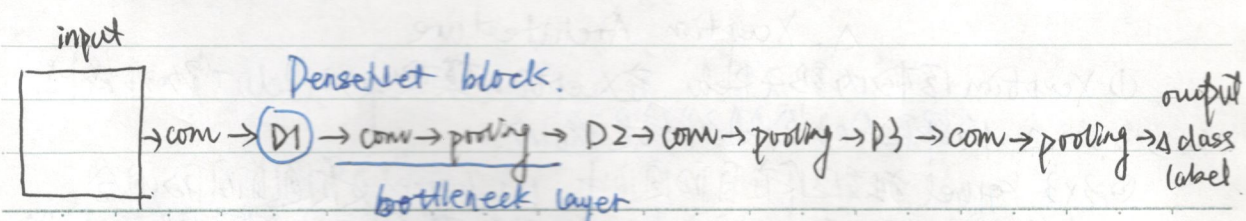
② Pyramid Network. 使用越来越宽的残差单元组合.

△. 全局路径 + 瓶颈层 + 残差单元

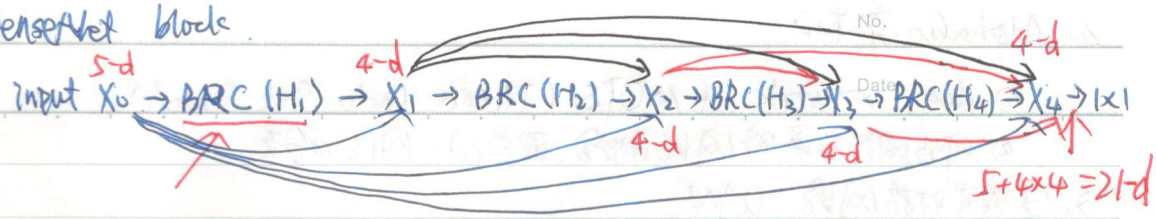



△ (M > N) 将M通道中的前N个通道加入N中. 效果优使用1x1卷积. 把N放大到M效果更佳.

③ DenseNet.



△. DenseNet block




BRC: BN (Batch_Normalization). Relu (Activation). Convolution layer

△. $X_{i+1} = H_{i+1}(X_i) + X_i$ 残差结构

DenseNet中体现3不同路径组合的思想，而ResNet中是不同路径的相加

△. DenseNet中通道增长很快，因此最后用1×1 conv缩小通道数

eg. ResNet: $X_4 = X_1 + H_1(X) + H_2(X + H_1(X)) + H_3(X + H_1(X)) + H_2(X_1 + H_1(X)))$

DesNet: $[X, H_1(X), H_2(X, H_1(X)), H_3(X, H_1(X), H_2(X, H_1(X)))]$

通道组合

从点对点加法变为通道组合，保留更多信息

△. 后续还有fractual Net. Dual Path Net等.

<9> 模型压缩

- squeezeNet. 参数压缩 采用低精度参数 △. 16位参数
- MobileNet. Inception结构
- shuffleNet. 通道组合 (XCeption) 过程中通道打散重组

△. 一般应用于移动端. PC端等

<10> 卷积核 (kernel) 的变形

- 扩张卷积 (dilated convolution) → 更大的感受野
- 可变形卷积 (deformable convolution) → 通过训练寻找3×3 kernel的最佳采样点

↑ 加入一个可训练的偏移层，为每个采样点找到合适的位置.

<11>. object detection.

- one stage YOLO系列. SSD系列.
- two stage. Faster R-CNN

△. mAP与FPS两个指标的演进.

segmentation.

- FCN.
- Mask R-CNN

△. 前沿技术

<12> style transfer

△. 重点在损失函数的设计. 也有用GAN方法实现的样例 (eg. Cycle GAN).

4. AlphaGo 架构

① deep network + MCTS (Monte Carlo Tree Search)

② 价值网络与策略网络融合，成为同一网络的分支。

5. 生成对抗网络：GAN

GAN：Generative Adversairal Networks

△. { 概率生成模型
判别模型

data (True) → D (Discrimator) → D(x) = 1

z 编码 → G (Generator) → G(z) → D → D(G(z)) = 0

对抗前行

△. algorithms：

① 选择真实数据 x，计算 D(x)，希望 D(x) = 1，获得反向梯度

② 随机采样生成 z (eg: z = {z1, ... zn} 其中 zi 为 std = 1 的正态分布)

③ 生成 G(z)，进而有 D(G(z))，希望 D(G(z)) = 0，获得反向梯度

④ 两次梯度相加，回传，从而训练 D

⑤ 将 G(z) 再次输入 D，得到新的 D(G(z))，此时希望 D(G(z)) = 1，获得相对于输入的梯度，反向回传训练 G

⑥ 迭代 ①-⑤ 演进 G, D

△. 训练过程中的优化目标：

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(x)}(1-D(G(z)))$$

进一步有：

$$\begin{cases} L_D = -E_{x\sim P_{data}(x)}[\log D(x)] - E_{z\sim P_z(x)}[\log(1-D(G(z)))] \\ L_G = -E_{z\sim P_z(x)}[\log(D(G(z)))] \end{cases}$$

其中 D：{ 希望 D(x) = 1，D(G(z)) = 0，则 $L_D = 0$
不希望 D(x) = 0，D(G(z)) = 1，则 $L_D = +\infty$

G { 希望 D(G(z)) = 1，则 $L_G = 0$
不希望 D(G(z)) = 0，则 $L_G = +\infty$

因此，D, G 的目标均为最小化各自的损失。

△. DCGAN：Deep Convolutional GAN 实例 → 潜在用于风格迁移 style transfer

核心思想：类别为X的图像x变到类别为Y的图像y，再变回类别为X的图像x'，则x与x'应尽可能相似。

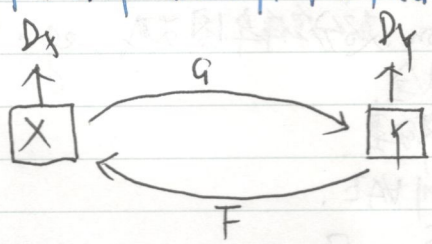
再论：

给定两个数据集X与Y.

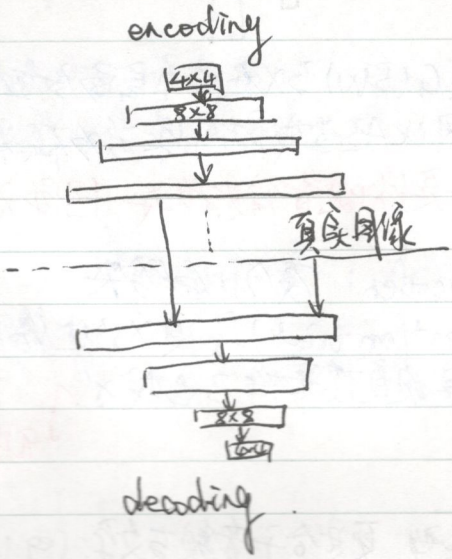
① G网络，x变为y，/ F网络，y变为x.

② $D_X$网络，判别F生成x的真假，$D_Y$网络，判别G生成y的真假.

③ 同时训练G、F、$D_X$、$D_Y$，希望F(G(x))与x相似，G(F(y))与y相似




<1>. 生成高分辨率图像：nVidia 论文 → 鲁棒可信赖，

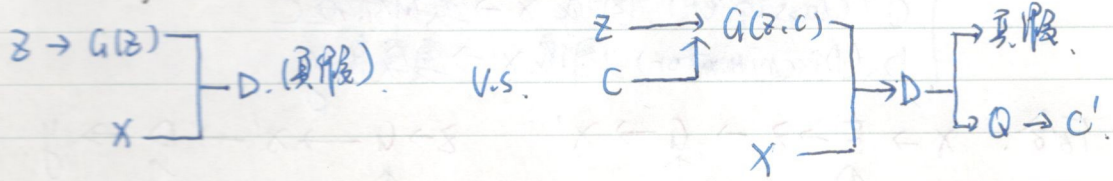



<2> 隐信息提取，InfoGAN.

在普通GAN的变码过程中加入了语义操作，将G的输入分为两部分c和z

c是符合所需格式的具有语义的隐变量（Latent Variable）

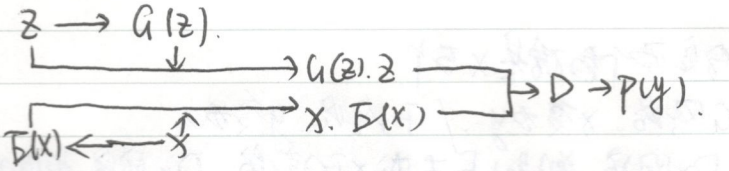
z是噪音，即与图像不联，但没有明确语义.




GAN.

<3>. GAN的其它应用

① ALI与BiGAN：




△. 即G学会从z生成x. E学会从x还原z.

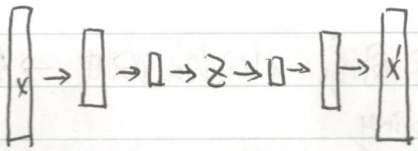
② super-resolution. 超分辨率图生成, eg: SRGAN.

③ 图像缺失部分补全.

④ 生成的部分非常逼真.

<4>. 自编码器：从AE到VAE.

(Auto-Encoder) AE.




△. 相比于GAN. AE的优化目标是G(E(x))与x在像素层面尽量相似. 而没有生成的判别网络D. 因此AE生成的图像较为模糊.

△. 这源自在信息压缩的过程中一定伴随着信息损失. 图中无法完全还原

△. VEA (Variational Auto-Encoder). 变分自编码器.
通过重参化技巧 (reparameterization trick) 实现x经过编码后的z符合某种特定分布. 从而实现直接采样z生成x' ↑ 无中生有.

<5>. PixelRNN与PixelCNN.
逐像素生成图像. 本质上一个序列模型. 更适合于音频与文字. (eg. WaveNet)

<6>. VAE与GAN的结合: CVAE-GAN.
C代表分类信息. 从而生成指定分类的图像.

四大组件：
- E (Encoder): 图像x → 编码z. 给定类别C 编码质量更高
- G (Generator). 编码z → 图像x. 给定C时生成指定类别图像
- C. (Classifier). 图像x → 类别标签C
- D. (Discriminator). 图像x → 真实度y.

结合: $x \to E \to z \to G \to x'$ (C → E, C → G)　　$z \to G \to x' \to D \to y$ (C → G, C → D)

CVAE架构　　CGAN架构

从效果看：
- CVAE：生成的图模糊，但均匀、光滑
- CGAN：生成的图清晰，但有明显瑕疵

因而到成立科，构建 CVAE-GAN 结构：

$$x \rightarrow \underset{\substack{\uparrow \\ c}}{E} \rightarrow z \rightarrow \underset{\substack{\uparrow \\ c}}{G} \rightarrow x' \begin{cases} C \rightarrow c \text{ (label)} \\ D \rightarrow y \text{ (confidence)} \end{cases}$$

其中G需有3个目标：
- ① 从 x 生成的 z，G要能还原出与接近的 x'，即AE的思想
- ② G生成的图像 x' 要能骗过判别器 D，即GAN的思想
- ③ G生成的图像 x' 要能被 C 判别为指定类别 c，即有infoGAN思想

△ 这里 c 代表不同类别，而 z 中应当包含其它的一切潜变特征，包括表情、面貌、光照等，从而仅变 c 即可在人脸当中保留人脸换换脸。

△ CVAE-GAN在语义插值上的良好表现，可用于 data Augmentation.

6. 通向智能之路..

△ adversarial examplex 对抗样本

深度网络擅长于精细分类，因为其对像素级导致的微小扰动运用远胜人类，即其更擅于细节。

△ Capsule 胶囊网络，gcForest 模型。