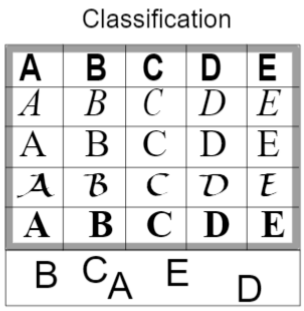
# 双线性

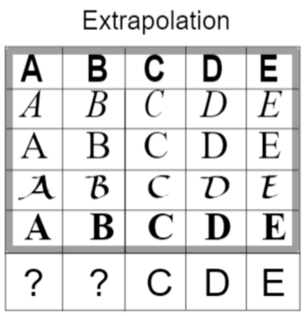
感知系统通常会将它们观察到的“内容（content）”因子和“风格（style）”因子分开，比如用“不熟悉的口音”说的“熟悉的单词”进行分类，识别字母时有不同字体或手写风格，或者识别在“不熟悉的观察”条件下看到的“熟悉的面部或物体"。这些任务和许多其他的基本的感知任务都有一个共同点，那就是需要分别处理两个独立的因子，这两个因子是一系列观察的基础。

具体列举*classification*，*extrapolation* 和 *translation* 三个例子如下，黑框内是训练数据，有不同的内容（字母）和不同的风格（字体）：

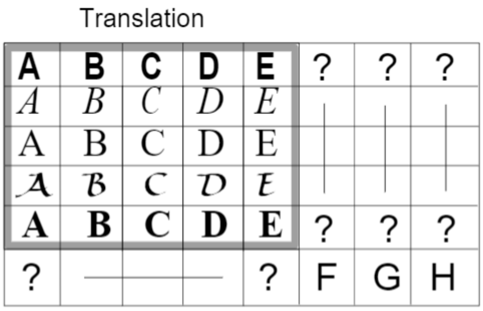
**Classification：** 对不同风格的内容进行分类




**Extrapolatio：** 推断内容在不同风格中的表现形式



**Translation：** 将仅在新风格中观察到的新内容转换为已知的风格或内容类。




对于上述的例子，如果模型可以发现所有数据中每一行独立于列、每一列独立于行 以及 与行列都独立 的共同点的参数化表示。那么就可以将不同参数结合，得到不同的表现形式（observation）（比如，不同字体下的各种字母）。

关于这方面的研究包括：主成分分析（principal component analysis）、独立成分分析（independent component analysis）、协同矢量量化（cooperative vector quantization）、层次因子模型（hierarchical factorial models）等等，本文则主要讲述**双线性模型**（bilinear models）[1]。

# 什么是双线性模型

双线性模型是具有**可分性**的双因子模型：当任意一个因子保持不变时，模型的输出对于另一个因子是线性的。根据标签是否对称分为对称模型和非对称模型。

- Linear

  假设$V, W$ 为线性空间，$f: V \rightarrow W$ 两个线性空间的映射，如果满足：

  $$f\left(v_{1}+v_{2}\right)=f\left(v_{1}\right)+f\left(v_{2}\right)$$
  $$f(\alpha v)=\alpha f(v)$$

  $f: V \rightarrow W$ 是线性的。

- Bilinear

  假设$U, V, W$ 为线性空间，$f: V \times U \rightarrow W$ ，如果满足：

  $$f\left(u_{1}+u_{2}, v\right)=f\left(u_{1}, v\right)+f\left(u_{2}, v\right)$$
  $$f\left(u, v_{1}+v_{2}\right)=f\left(u, v_{1}\right)+f\left(u, v_{2}\right)$$
  $$f(\alpha u, v)=\alpha f(u, v)=f(u, \alpha v)$$

  $f: V \times U \rightarrow W$ 是双线性的。

  当 $v$ 固定，$f(u, v)$ 在 $u$ 中是线性的：

  $$\begin{aligned} &f(u, v)=f_{v}(u)=f_{v}\left(u_{1}+u_{2}\right)=f_{v}\left(u_{1}\right)+f_{v}\left(u_{2}\right) \\ &f(\alpha u, v)=f_{v}(\alpha u)=\alpha f_{v}(u) \end{aligned}$$

- 张量积

  $$\mathbf{b} \otimes \mathbf{a} \rightarrow\left[\begin{array}{l} b_{1} \\ b_{2} \\ b_{3} \\ b_{4} \end{array}\right]\left[\begin{array}{lll} a_{1} & a_{2} & a_{3} \end{array}\right]=\left[\begin{array}{lll} a_{1} b_{1} & a_{2} b_{1} & a_{3} b_{1} \\ a_{1} b_{2} & a_{2} b_{2} & a_{3} b_{2} \\ a_{1} b_{3} & a_{2} b_{3} & a_{3} b_{3} \\ a_{1} b_{4} & a_{2} b_{4} & a_{3} b_{4} \end{array}\right]$$

  结果的维数为 4×3 = 12， 结果的秩为1。

## 对称模型

对称模型中，用向量 $\mathbf{a}^{s}$ 和 $\mathbf{b}^{c}$ 来表示style和content，分别有 $I$ 和 $J$ 维，用 $k$ 维的 $y^{sc}$ 来表示样式s下c的观察向量。那么

$$y_k^{sc} = \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ijk} a_i^s b_j^c$$

其中 $i, j, k$ 表示style、content和observation的组成部分，$w$ 独立于style和content，表示这两个因子之间的相互作用。用向量的形式，可以写为：

$$y_k^{sc} = \mathbf{a}^{s\mathrm{T}} \mathbf{W}_k \mathbf{b}^c$$

**K个矩阵 $\mathrm{W}_k$ 描述了从style和content空间到K维observation空间的双线性映射。**

除此之外也可以写成如下形式：

$$\mathrm{y}^{sc} = \sum_{i,j} \mathbf{w}_{ij} a_i^s b_j^c$$

**$\mathbf{w}_{ij}$ 是一个k维的向量，那么 $\mathrm{y}^{sc}$ 可以看作是由这些基向量混合$\mathrm{a}^s$和$\mathrm{b}^c$的张量积得到。**

# 非对称模型

有时，在训练中学习的一些基本style的线性组合可能不能很好地描述新的style。可以通过让交互项 $w_{ijk}$ 本身随style变化来获得更灵活的不对称模型。

$$y_k^{sc} = \sum_{i,j} w_{ijk}^s a_i^s b_j^c$$

讲上式的style相关项合并：

$$\begin{aligned} a_{jk}^s &= \sum_i w_{ijk}^s a_i^s \\ y_k^{sc} &= \sum_j a_{jk}^s b_j^c \end{aligned}$$

用 $\mathbf{A}^s$ 来表示由分量 $\{a_{jk}^s\}$ 组成的 $I \times K$ 矩阵：

$$\mathbf{y}^{\mathrm{sc}} = \mathbf{A}^s \mathbf{b}^c$$

**这里 $a_{jk}^s$ 可以看作从content space到observation space特定于style的映射。**

让 $\mathbf{a}_j^{\mathrm{s}}$ 表示含有分量 $\{a_{jk}^s\}$ 的k维向量，则式子可以写为：

$$\mathrm{y}^{\mathrm{sc}} = \sum_j \mathbf{a}_j^{\mathrm{s}} b_j^{\mathrm{c}}$$

可以认为 $a_{jk}^s$ 是一组特定于style的基向量，这些基向量和特定于content的系数 $b_j^{\mathrm{c}}$ 混合在一起后产生observation向量。

# 模型应用

## 单模态

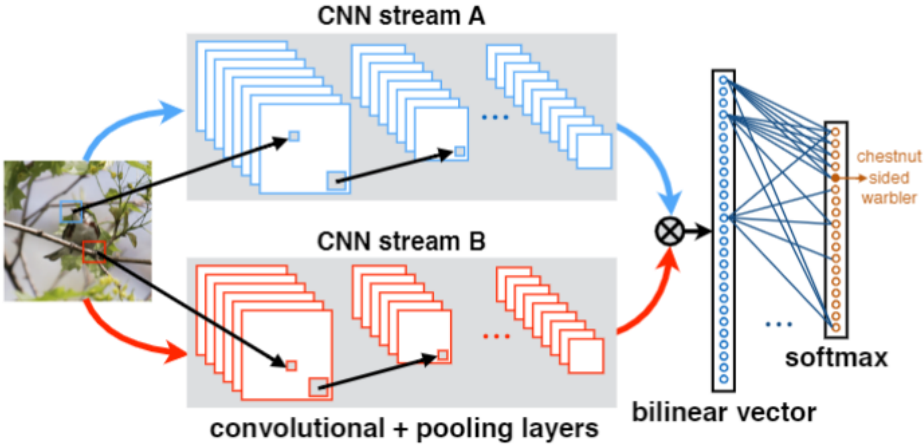
### Second-Order Pooling[2]

文章提出了二阶池化的方法：

$$\begin{aligned} \mathbf{G}_{avg}\left(R_j\right) &= \frac{1}{\left|F_{R_j}\right|} \sum_{i:\left(\mathbf{f}_i \in R_j\right)} \mathbf{x}_i \cdot \mathbf{x}_i^{\top} \\ \mathbf{G}_{\max}\left(R_j\right) &= \max_{i:\left(\mathbf{f}_i \in R_j\right)} \mathbf{x}_i \cdot \mathbf{x}_i^{\top} \end{aligned}$$

$\mathbf{x}_i \in \mathbb{R}^n$ 是位置 $i$ 处的局部特征，$|F_{R_j}|$ 是在区域$R_j$范围内的局部特征数目。文章发现使用了二阶的信息会比使用一阶的信息性能更为优秀。

## Bilinear CNN[3]




双线性映射 $f: V \times U \rightarrow W$ 中，$V$ 和 $U$ 可以用来表示不同的信息，文章想通过two-stream结构来分别学习到位置（location）信息和图像（image）信息。

$$\text{bilinear } (l, I, f_A, f_B) = f_A(l, I)^T f_B(l, I)$$

其中特征函数为 $f: \mathcal{L} \times \mathcal{I} \rightarrow \mathbb{R}^{K \times D}$，把图片 $I \in \mathcal{I}$ 和 位置 $l \in \mathcal{L}$ 作为输入，输出 $K \times D$ 的特征（文章选择了预训练的CNN在中间层截断作为特征函数）。

$$\begin{gathered} f_A(l, I) \in \mathbb{R}^{K \times D_A} \\ f_B(l, I) \in \mathbb{R}^{K \times D_B} \\ f_A(l, I)^T f_B(l, I) \in \mathbb{R}^{D_A \times D_B} \\ \Phi(I) = \sum_{l \in \mathcal{L}} \text{ bilinear } (l, I, f_A, f_B) = \sum_{l \in \mathcal{L}} f_A(l, I)^T f_B(l, I) \end{gathered}$$

然后对每个位置上的特征进行求和得到全局的特征，将其展开成向量后使用线性变换降维，经过signed square-root和归一化操作后做分类预测。

> 1. Bilinear CNN 的想法是想用两个CNN分别学到位置信息和图像信息，但由于CNN是一个黑盒，所以实际上并不能说两个CNN学到的就是想让它们学到的信息，但确实是检测出不同的信息，而提升了性能。
> 2. 上述两篇的文章非常相似，Second-Order Pooling 相当于$K = 1$，并且$f_A = f_B$ 的 Bilinear CNN，如果$f_A$ 和 $f_B$ 是相同的，也可以叫做同源双线性池化（Homogeneous Bilinear Pooling）。

## Compact Bilinear Pooling[4]

$$B(\mathcal{X}) = \sum_{s \in \mathcal{S}} x_s x_s^T$$

Bilinear的运算使得特征的维度变成了原来的平方，这增加后续的计算量，这篇文章提出一种紧凑的双线性池化。

原始特征经过双线性池化后用于最后的分类预测，如果从核方法的角度来，那么双线性可以写成如下形式：

$$\begin{aligned}\langle B(\mathcal{X}), B(\mathcal{Y})\rangle &= \left\langle \sum_{s\in\mathcal{S}} x_s x_s^T, \sum_{u\in\mathcal{U}} y_u y_u^T \right\rangle \\ &= \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} \left\langle x_s x_s^T, y_u y_u^T \right\rangle \\ &= \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} \langle x_s, y_u\rangle^2\end{aligned}$$

如果能够找到一个低维映射函数 $\phi(x) \in R^d$， 其中 $d << c^2$，且满足 $\langle\phi(x), \phi(y)\rangle \approx k(x, y) = \langle x_s, y_u\rangle^2$，那么上式就可以写成：

$$\begin{aligned}\langle B(\mathcal{X}), B(\mathcal{Y})\rangle &= \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} \langle x_s, y_u\rangle^2 \\ &\approx \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}} \langle\phi(x), \phi(y)\rangle \\ &\equiv \langle C(\mathcal{X}), C(\mathcal{Y})\rangle\end{aligned}$$

文章基于这个思想提出了 Random Maclaurin (RM) 和 Tensor Sketch (TS) 两种方法

- Random Maclaurin (RM)：RM是一种显式低维特征映射来近似多项式核的方法

  **Algorithm 1** Random Maclaurin Projection

  Input: $x \in \mathbb{R}^c$
  Output: feature map $\phi_{RM}(x) \in \mathbb{R}^d$, such that $\langle\phi_{RM}(x), \phi_{RM}(y)\rangle \approx \langle x, y\rangle^2$
  1. Generate random but fixed $W_1, W_2 \in \mathbb{R}^{d\times c}$, where each entry is either $+1$ or $-1$ with equal probability.
  2. Let $\phi_{RM}(x) \equiv \frac{1}{\sqrt{d}}(W_1 x) \circ (W_2 x)$, where $\circ$ denotes element-wise multiplication.

  如果 $w_1, w_2 \in \mathbb{R}^c$ 是随机的 $\{-1, +1\}$ 向量，且 $\phi(x) = \langle w_1, x\rangle \langle w_2, x\rangle$，对于非随机的 $x, y \in \mathbb{R}^c$，有 $E[\phi(x)\phi(y)] = E[\langle w_1, x\rangle \langle w_1, y\rangle]^2 = \langle x, y\rangle^2$。因此RM中每个投影项都有一个近似的期望。

- Tensor Sketch (TS)： TS使用sketch function[5]，这个函数有属性：$E[\langle\Psi(x, h, s), \Psi(y, h, s)\rangle] = \langle x, y\rangle$，除此之外 $\Psi(x \otimes y, h, s) = \Psi(x, h, s) * \Psi(y, h, s)$ [6]

  **Algorithm 2** Tensor Sketch Projection

  Input: $x \in \mathbb{R}^c$
  Output: feature map $\phi_{TS}(x) \in \mathbb{R}^d$, such that $\langle\phi_{TS}(x), \phi_{TS}(y)\rangle \approx \langle x, y\rangle^2$
  1. Generate random but fixed $h_k \in \mathbb{N}^c$ and $s_k \in \{+1, -1\}^c$ where $h_k(i)$ is uniformly drawn from $\{1, 2, \ldots, d\}$, $s_k(i)$ is uniformly drawn from $\{+1, -1\}$, and $k = 1, 2$.
  2. Next, define sketch function $\Psi(x, h, s) = \{(Qx)_1, \ldots, (Qx)_d\}$, where $(Qx)_j = \sum_{t:h(t)=j} s(t)x_t$
  3. Finally, define $\phi_{TS}(x) \equiv \mathrm{FFT}^{-1}(\mathrm{FFT}(\Psi(x, h_1, s_1)) \circ \mathrm{FFT}(\Psi(x, h_2, s_2)))$, where the $\circ$ denotes element-wise multiplication.

  > sketch function 用于数据流中高频项的技术，可以用在这里是因为数学性质的匹配，以及可使用FFT优化，加速运算。同理也可使用其他拥有合适数学性质的函数代替。

## Low-rank Bilinear Pooling[7]

前面的双线性融合方法，$\mathbf{X} \in \mathbb{R}^{c\times hw}$ 经过双线性池化得到双线性特征后，展开成向量 $\mathbf{z} = \mathrm{vec}\left(\mathbf{X}\mathbf{X}^T\right) \in \mathbb{R}^{c^2}$，使用线性分类器做最后的预测。假设是一个用 $\mathbf{w} \in \mathbb{R}^{c^2}$ 和 $b$ 参数化的线性分类器，标准的soft margin SVM目标函数为：

$$\min_{\mathbf{w},b} \frac{1}{N}\sum_{i=1}^{N}\max\left(0,1-y_i\mathbf{w}^T\mathbf{z}_i+b\right)+\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

上式写成矩阵的形式可以表示为：

$$\min_{\mathbf{W},b} \frac{1}{N}\sum_{i=1}^{N}\max\left(0,1-y_i\operatorname{tr}\left(\mathbf{W}^T\mathbf{X}_i\mathbf{X}_i^T\right)+b\right)+\frac{\lambda}{2}\|\mathbf{W}\|_F^2$$

利用拉格朗日函数对参数求偏导，可以得到上述两个式子的最优解为：

$$\begin{gathered}\mathbf{w}^*=\sum_{y_i=1}\alpha_i\mathbf{z}_i-\sum_{y_i=-1}\alpha_i\mathbf{z}_i\\ \mathbf{W}^*=\sum_{y_i=1}\alpha_i\mathbf{X}_i\mathbf{X}_i^T-\sum_{y_i=-1}\alpha_i\mathbf{X}_i\mathbf{X}_i^T\\ \text{where }\alpha_i\geq 0,\forall i=1,\ldots,N\end{gathered}$$

因为 $\mathbf{z}=\operatorname{vec}\left(\mathbf{X}\mathbf{X}^T\right)\in\mathbb{R}^{c^2}$，所以$\mathbf{w}^*=\operatorname{vec}(\mathbf{W}^*)$。 $\mathbf{W}^*$ 是对称矩阵的求和，因此 $\mathbf{W}^*$ 也是对称矩阵。从上式看出 $\mathbf{W}^*$ 是正样本和负样本分别对应的矩阵的差值，所以可以做以下特征值分解：

$$\begin{aligned}\mathbf{W}^*&=\mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Psi}^T=\mathbf{\Psi}_+\mathbf{\Sigma}_+\mathbf{\Psi}_+^T+\mathbf{\Psi}_-\mathbf{\Sigma}_-\mathbf{\Psi}_-^T\\ &=\mathbf{\Psi}_+\mathbf{\Sigma}_+\mathbf{\Psi}_+^T-\mathbf{\Psi}_-\left|\mathbf{\Sigma}_-\right|\mathbf{\Psi}_-^T\\ &=\mathbf{U}_+\mathbf{U}_+^T-\mathbf{U}_-\mathbf{U}_-^T\end{aligned}$$

其中 $\mathbf{\Sigma}_+$ 和 $\mathbf{\Sigma}_-$ 分别是正值和负值的特征值， $\mathbf{\Psi}_+$ 和 $\mathbf{\Psi}_-$ 为对应的特征向量。 第三行的 $\mathbf{U}_+=\mathbf{\Psi}_+\mathbf{\Sigma}_+^{\frac{1}{2}}$ 以及 $\mathbf{U}_-=\mathbf{\Psi}_-|\mathbf{\Sigma}_-|^{\frac{1}{2}}$。因此 $\mathbf{W}^*$ 可能会有好的低秩的分解，即 $\mathbf{U}_-$ 和 $\mathbf{U}_+$ 是低秩的。文章直接施加了一个强硬的低秩约束 $\operatorname{rank}(\mathbf{W})=r\ll c$，具体的方法是使用 $\mathbf{U}_+,\mathbf{U}_-\in\mathbb{R}^{c\times r/2}$ 来近似表示 $\mathbf{W}$：

$$\begin{aligned}y&=\operatorname{tr}\left(\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^T\right)+b\\ &=\operatorname{tr}\left(\boldsymbol{U}_+\boldsymbol{U}_+^T-\boldsymbol{U}_-\boldsymbol{U}_-^T\right)\boldsymbol{X}\boldsymbol{X}^T+b\\ &=\operatorname{tr}\left(\boldsymbol{U}_+\boldsymbol{U}_+^T\boldsymbol{X}\boldsymbol{X}^T\right)-\operatorname{tr}\left(\boldsymbol{U}_-\boldsymbol{U}_-^T\boldsymbol{X}\boldsymbol{X}^T\right)+b\\ &=\operatorname{tr}\left(\left(\boldsymbol{U}_+^T\boldsymbol{X}\right)^T\left(\boldsymbol{U}_+^T\boldsymbol{X}\right)\right)-\operatorname{tr}\left(\left(\boldsymbol{U}_-^T\boldsymbol{X}\right)^T\left(\boldsymbol{U}_-^T\boldsymbol{X}\right)\right)+b\\ &=\left\|\boldsymbol{U}_+^T\boldsymbol{X}\right\|_{\mathrm{F}}^2-\left\|\boldsymbol{U}_-^T\boldsymbol{X}\right\|_{\mathrm{F}}^2+b\end{aligned}$$

因此不需要计算$\mathbf{X}\mathbf{X}^T$ ，而降低了计算量。

## Factorized Bilinear Model[8]

根据前面介绍的双线性池化，双线性池化结合全连接层的模型可以写为：

$$\begin{aligned}y_j&=b_j+\mathbf{W}_{j\cdot}^T\operatorname{vec}\left(\sum_{i\in\mathbb{S}}\mathbf{x}_i\mathbf{x}_i^T\right)\\ &=b_j+\sum_{i\in\mathbb{S}}\mathbf{x}_i^T\mathbf{W}_{j\cdot}^R\mathbf{x}_i\end{aligned}$$

其中 $\mathbf{W}_{j\cdot}$ 是 $\mathbf{W}$ 的第 $j$ 行，$\mathbf{W}_{j\cdot}^R\in\mathbb{R}^{n\times n}$ 是 $\mathbf{W}_{j\cdot}$ reshape 之后的矩阵，$y_j$ 和 $b_j$ 分别是 $\mathbf{y}$ 和 $\mathbf{b}$ 第 $j$ 个值。因此根据这个形式提出了结合一阶和二阶特征，并且使用低秩矩阵 $\mathbf{F}$ 来代替 $\mathbf{W}$ 简化计算的模型：

$$y=b+\mathbf{w}^T\mathbf{x}+\mathbf{x}^T\mathbf{F}^T\mathbf{F}\mathbf{x}$$

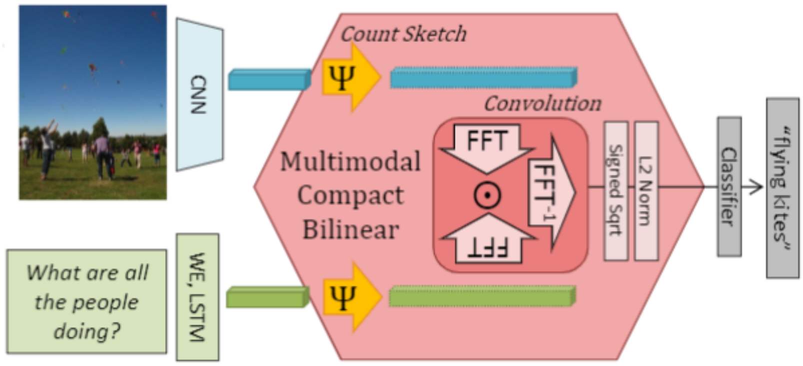
$\mathbf{x}\in\mathbb{R}^n$ 是输入向量，$\mathbf{w}\in\mathbb{R}^n$ 是权重， $\mathbf{F}\in\mathbb{R}^{k\times n}$ 表示与 k 个因子的交互权重，矩阵表达式可以解释为：

$$y=b+\sum_{i=1}^{n} w_i x_i+\sum_{i=1}^{n}\sum_{j=1}^{n}\left\langle \mathbf{f}_{\cdot i}, \mathbf{f}_{\cdot j}\right\rangle x_i x_j$$

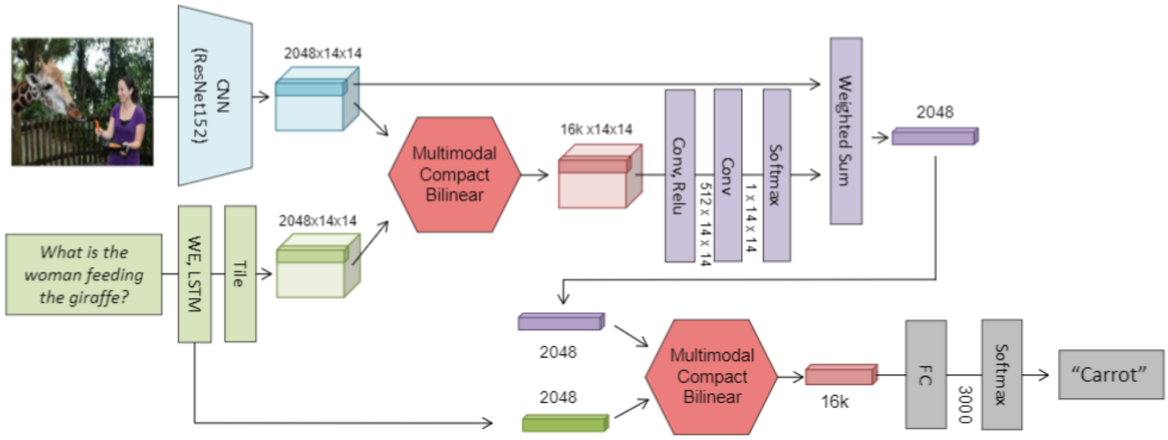
$\mathbf{f}_{\cdot i}$ 和 $\mathbf{f}_{\cdot j}$ 是 $\mathbf{F}$ 的第 $i$ 行和第 $j$ 行，$\langle \mathbf{f}_{\cdot i}, \mathbf{f}_{\cdot j}\rangle$ 表示输入特征向量第 $i$ 和第 $j$ 个值的交互。

## 多模态

### Multimodal Compact Bilinear Pooling[9]




文章将Compact Bilinear Pooling中TS的方法，套用到VQA上，构建了如上的Multimodal Compact Bilinear Module。整体网络结构使用了两次MCB模块，第一次用来计算图像的注意力分布，第二次将图像信息与文本信息融合：




### Multimodal Low-rank Bilinear Pooling[10]

$$f_i=\sum_{j=1}^{N}\sum_{k=1}^{M} w_{ijk} x_j y_k+b_i=\mathbf{x}^T \mathbf{W}_i \mathbf{y}+b$$

还是一样的双线性模型，$\mathbf{W}_i \in \mathbb{R}^{N\times M}$ 是 $f_i$ 对应的参数，共有 $L$ 个，那么参数一共有 $L\times(N\times M+1)$ 个，引入前面提到的低秩分解，降低计算量：

$$f_i=\mathbf{x}^T \mathbf{W}_i \mathbf{y}+b_i=\mathbf{x}^T \mathbf{U}_i \mathbf{V}_i^T \mathbf{y}+b_i=\mathbb{1}^T\left(\mathbf{U}_i^T \mathbf{x}\circ \mathbf{V}_i^T \mathbf{y}\right)+b_i$$

$\mathbb{1}\in\mathbb{R}^d$ 表示为1的列向量，$\circ$ 表示哈达玛积，但仍然需要两个三阶张量 $\mathbf{U}$ 和 $\mathbf{V}$，为了减少参数张量的阶数，用 $\mathbb{P}\in\mathbb{R}^{d\times c}$ 来代替 $\mathbb{1}$：

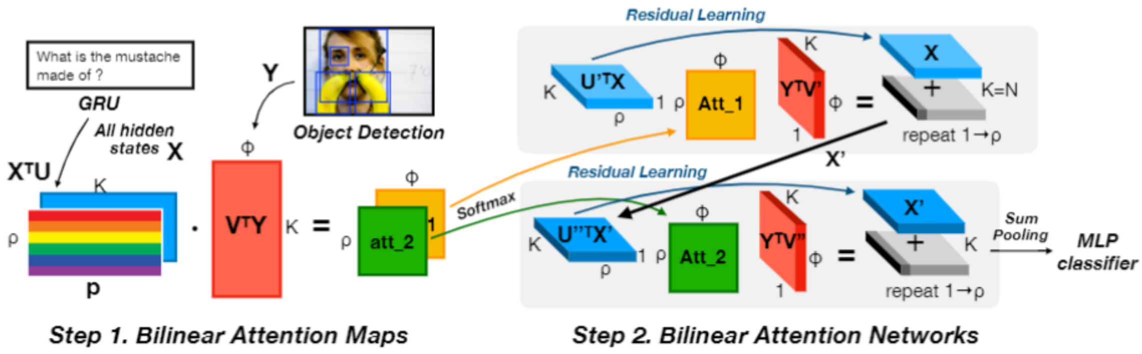
$$\mathbf{f}=\mathbf{P}^T\left(\mathbf{U}^T \mathbf{x}\circ \mathbf{V}^T \mathbf{y}\right)+\mathbf{b}$$

$\mathbf{U}$ 和 $\mathbf{V}$ 也可以有自己的偏置项，因此完整的模型为：

$$\begin{aligned}\mathbf{f}&=\mathbf{P}^T\left(\left(\mathbf{U}^T \mathbf{x}+\mathbf{b}_x\right)\circ\left(\mathbf{V}^T \mathbf{y}+\mathbf{b}_y\right)\right)+\mathbf{b}\\&=\mathbf{P}^T\left(\mathbf{U}^T \mathbf{x}\circ \mathbf{V}^T \mathbf{y}+\mathbf{U}'^T \mathbf{x}+\mathbf{V}'^T \mathbf{y}\right)+\mathbf{b}'\end{aligned}$$

其中，$\mathbf{U}'^T=\operatorname{diag}(\mathbf{b}_y)\cdot\mathbf{U}^T$ ，$\mathbf{V}^T=\operatorname{diag}(\mathbf{b}_x)\cdot\mathbf{V}^T$ , $\mathbf{b}'=\mathbf{b}+\mathbf{P}^T\left(\mathbf{b}_x\circ\mathbf{b}_y\right)$

### Bilinear Attention Networks[11]




在双线性的基础上引入注意力，通过 softmax 来得到对多通道输入 $\mathbf{Y}$ 各个通道的注意力分布：

$$\alpha:=\operatorname{softmax}\left(\mathbf{P}^T\left(\left(\mathbf{U}^T\mathbf{x}\cdot 1^T\right)\circ\left(\mathbf{V}^T\mathbf{Y}\right)\right)\right)$$

将上式扩展到两个输入都为多通道，即考虑两两通道之间的注意力，$\mathcal{A}\in\mathbb{R}^{\rho\times\phi}$：

$$\mathcal{A}:=\operatorname{softmax}\left(\left(\left(1\cdot\mathbf{p}^T\right)\circ\mathbf{X}^T\mathbf{U}\right)\mathbf{V}^T\mathbf{Y}\right)$$

注意到经过 softmax 之前的每一个元素，是双线性融合的输出：

$$A_{i,j}=\mathbf{p}^T\left(\left(\mathbf{U}^T\mathbf{X}_i\right)\circ\left(\mathbf{V}^T\mathbf{Y}_j\right)\right)$$

可以拓展多重注意力：

$$\mathcal{A}_g:=\operatorname{softmax}\left(\left(\left(1\cdot\mathbf{p}_g^T\right)\circ\mathbf{X}^T\mathbf{U}\right)\mathbf{V}^T\mathbf{Y}\right)$$

通过注意力分布可以融合两个输入：

$$\mathbf{f}_k'=\left(\mathbf{X}^T\mathbf{U}'\right)_k^T\mathcal{A}\left(\mathbf{Y}^T\mathbf{V}'\right)_k$$

因为上式也是双线性的形式，所以可以被写成：

$$\mathbf{f}_k'=\sum_{i=1}^{\rho}\sum_{j=1}^{\phi}\mathcal{A}_{i,j}\left(\mathbf{X}_i^T\mathbf{U}_k'\right)\left(\mathbf{V}_k'^T\mathbf{Y}_j\right)=\sum_{i=1}^{\rho}\sum_{j=1}^{\phi}\mathcal{A}_{i,j}\mathbf{X}_i^T\left(\mathbf{U}_k'\mathbf{V}_k'^T\right)\mathbf{Y}_j$$

> Bilinear attention 用双线性特征求出注意力分布，随后用双线性池化进行融合

## 参考文献


[1] Separating Style and Content with Bilinear Models

[2] Semantic Segmentation with Second-Order Pooling

[3] Bilinear CNNs for Fine-grained Visual Recognition

[4] Compact Bilinear Pooling

[5] Finding frequent items in data streams

[6] Fast and scalable polynomial kernels via explicit feature maps

[7] Low-rank Bilinear Pooling for Fine-Grained Classification

[8] Factorized Bilinear Models for Image Recognition

[9] Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding

[10] Hadamard product for low-rank bilinear pooling

[11] Bilinear Attention Networks