这是一份关于**“注意力机制与Transformer”**的机器学习复习笔记，基于你提供的PDF内容进行了逐页整理与要点扩充。

第一部分：词向量（Word Embedding)

本部分主要解决如何将非结构化的文字转化为计算机可处理的数值向量。

- **P4：词向量定义与优势**
  - **定义**：一种学习到的文本表示形式，语义相似的词在向量空间中距离更近。
  - **优势**：
    1. **计算友好**：神经网络工具箱更擅长处理稠密、低维向量，而非高维稀疏向量（如One-hot）。
    2. **泛化能力**：密集的表示有助于模型理解词与词之间的联系。
- **P5-P6：词袋模型（Bag of Words, BOW）**
  - **原理**：统计每个词出现的次数，忽略语法和词序，将其转为固定长度的向量。
  - **局限**：无法体现语义关系（例如“猫追狗”和“狗追猫”的向量可能一致），且存在维度灾难。
- **P7-P9：Word2Vec（经典统计方法）**
  - **CBOW (Continuous Bag of Words)**：根据上下文窗口词（Context）预测中心词（Target）。
  - **Skip-gram**：根据中心词预测上下文窗口词。
  - **目标函数**：最大化平均对数概率。使用 **Softmax** 计算概率分布。
  - **核心特性（P9）**：具有**线性平移特性**（Vector Arithmetic）。例如：`vec("Madrid") - vec("Spain") + vec("France") ≈ vec("Paris")`。这证明了向量空间捕捉到了语义逻辑（国家与首都的关系）。

第二部分：Transformer 与 注意力机制

本部分是现代大模型（如GPT, BERT）的核心。

- **P11-P13：注意力机制（Attention）的引入**
  - **动机**：解决歧义性。如 "bank" 在 "river bank"（河岸）和 "bank cash"（银行现金）中含义不同，Attention让模型根据上下文自动赋予不同权重。
  - **Bahdanau Attention（对齐与翻译）**：在机器翻译中，生成目标词时应重点“注视”源句子中的特定词。

- **P14-P15：编解码器中的注意力公式**
  - **上下文向量** c_i：输入序列隐藏状态 h_j 的加权总和。
  - **权重** α_{ij}：通过 Softmax 对对齐得分进行归一化得到。
  - **双向表示**：通过拼接正向和反向的隐状态 $h = [\overrightarrow{h}; \overleftarrow{h}]$ 来获得更丰富的词义。
- **P16-P18：Transformer 架构核心（"Attention is All You Need"）**
  - **基础构成**：$N = 6$ 层相同的编码器和解码器。包含：多头自注意力（Multi-head Self-attention）、前馈神经网络（FFN）、残差连接（Add）和层归一化（Norm）。
  - **缩放点积注意力（Scaled Dot-Product Attention）**：
    - 公式：$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
    - **Q (Query)**：搜索信息。
    - **K (Key)**：匹配信息。
    - **V (Value)**：内容信息。
    - $\sqrt{d_k}$ **的作用**：防止点积结果过大导致 Softmax 梯度消失。
- **P19：多头注意力（Multi-Head Attention）**
  - **意义**：允许模型在不同的代表子空间（subspaces）同时关注不同位置的信息。
- **P20：前馈网络（FFN）与嵌入**
  - 公式：$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$（即：线性层 + ReLU + 线性层）。
- **P21：位置编码（Positional Encoding）**
  - **必要性**：Transformer 本身是并行的，不像 RNN 具有时序性。为了让模型知道词的顺序，必须注入位置信息。
  - **方法**：使用不同频率的 Sine 和 Cosine 函数生成位置向量。

第三部分：视觉 Transformer (Vision Transformers, ViT)

本部分展示了 Transformer 如何从 NLP 跨界到 CV。

- **P23：ViT (Vision Transformer)**
  - **操作流**：将图像切分为固定大小的 **Patches**（补丁） -> 展平后进行线性投影 -> 加入位置嵌入 -> 输入标准 Transformer 编码器。
  - **[class] token**：额外添加一个可学习的标签位，用于最后的图像分类。

- **P24-P28：Swin Transformer（层次化视觉 Transformer）**
  - **改进点**：
    1. **层次化构建**：像 CNN 一样通过 Patch Merging 逐渐减小特征图尺寸，捕捉多尺度特征。
    2. **移动窗口自注意力（Shifted Window MSA）**：将注意力计算限制在局部窗口内，将计算复杂度从图像大小的**平方级**降低到**线性级**。
  - **性能**：Swin-B 在 ImageNet 上比同级别的 ViT 准确率更高，且推理速度更快。
- **P29-P32：DETR (DEtection TRansformer)**
  - **定义**：首个将 Transformer 用于目标检测的方法。
  - **特点**：
    1. 使用 CNN 提取特征 + Transformer 编解码器预测边界框。
    2. **Object Queries**：一组学习到的位置查询向量。
    3. **双边匹配损失（Bipartite matching loss）**：直接预测集合，**不需要 NMS（非极大值抑制）** 后处理。
- **P33：BEVFormer v2（前沿应用：自动驾驶）**
  - **核心**：将多视图（Multi-view）图像转换为鸟瞰图（Bird's-Eye-View, BEV）进行感知。
  - **结构**：结合了空间编码（Spatial Encoder）处理多相机融合，以及时间编码（Temporal Encoder）处理历史帧序列。

💡 复习重点建议：

1. **必背公式**：缩放点积注意力公式（P17）、位置编码原理（P21）。
2. **对比理解**：
   - **ViT vs Swin**：ViT 是全局注意力（计算量大），Swin 是窗口局部注意力 + 移动窗口（计算高效、层次化）。
   - **RNN vs Transformer**：RNN 串行计算（慢）、长距离依赖差；Transformer 并行计算（快）、长距离捕捉强（靠 Attention）。
3. **计算复杂度**：理解为什么 Swin Transformer 要引入 Window 概念（为了处理高分辨率图像时的计算效率问题）。