

IREEDGE

1.概述

IREEDGE是利用深度学习模型解决全芯片压降预测的问题。这个模型的主要做法是实现一个图到图的翻译任务，输入的是一个电流图，经过图像编码器、引擎、图像解码器，最后得到一个压降的热力图。

IREEDGE这篇论文很明显是Powernet出现之后的工作，因为看起来它似乎总是在与Powernet做比较。

论文中说，它们的模型与传统的CNN有所不同，他们不会输出标量，而是设置了一个解码器，把编码器中卷积操作逐渐减小的维度再扩充回来，直接得到一张热力图。

此外，模型中不涉及任何与输入图像尺寸有关的组件。它们的编码器虽然是卷积层，但实际上仅仅存储了卷积核的参数；也就是说，不论输入的图像有多大，它总是按部就班地拿卷积核在图片上扫。

最后，它们是直接基于整块芯片做静态压降预测，这跟之前Powernet的分tile训练不同。这样做的好处是不用再费劲寻找合适的tile size了。找tile size可以说是powernet中一个主要的超参数调优问题。

2.问题建立

基于数据库中给定的特征集合，我们要输出一张全芯片压降的热力图。

PDN: "Power Delivery Network" (电源分配网络) 的缩写。在集成电路设计中，PDN是用于将电源电压传递给芯片内各个功能区域和组件的网络结构。PDN的主要功能是提供稳定的电源电压和电流，以满足芯片内各个电路和器件的电源需求。通常由电源（如电源引脚或电源焊盘power pad）、电源线路、电源平面和电源滤波器等组成。

在实际面对问题的时候，我们应该获取这三个数据库:

- Power database:包含Internal power,Leakage power,Switching power,Activity factors里面大部分都跟Powernet是相同的。
- Layout database:包含block locations,cell locations,Block sizes,PDN stripes.
- Package Information:Power pad locations.

参数	说明
Internal power	器件内部的总功耗
Leakage power	器件的静态功耗,即无信号输入时的功耗
Switching power	器件的动态功耗,即器件在信号切换时的功耗
Activity factors	活动因子,表示器件内各部分的切换活动度,用于估算动态功耗
block locations	芯片上各个功能块的坐标位置
cell locations	芯片上每个标准单元的坐标位置
Block sizes	每个功能块的长宽尺寸
PDN stripes	为芯片提供电源的金属配线条的位置和尺寸
Power pad locations	芯片与外部电源连接的焊盘的坐标位置

在收集到这三个数据库中的这些参数之后，我们可以针对每一个数据库生成一张图：Power database 生成Power map(这次不是powernet中的tile而是整个芯片),Layout Information+Power database 生成PDN density map(反映PDN电路元件密度分布的图)，Package Information+ Layout Information 生成Effective distance to pad.如下图所示。

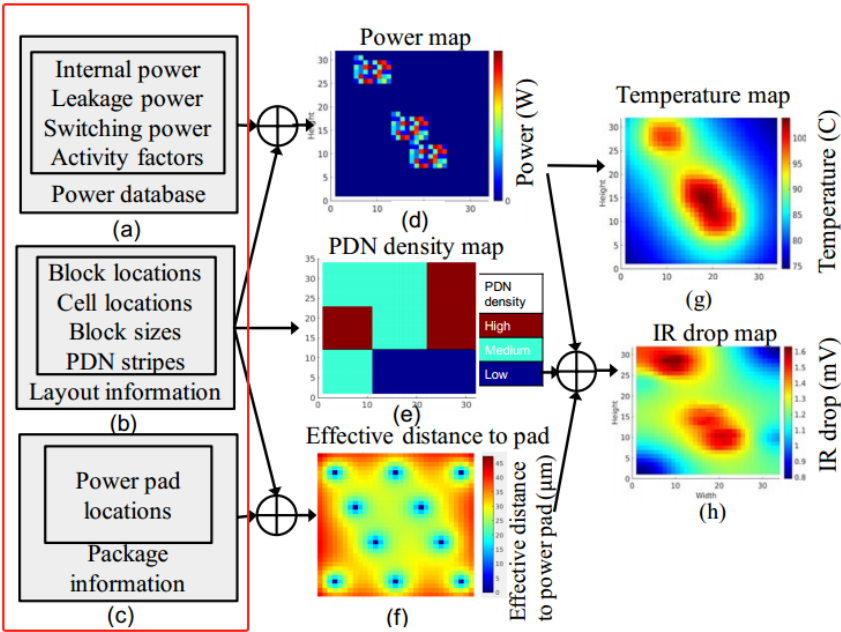


Fig. 2. Data representation: Mapping PDN and thermal analysis problems into image-to-image translations tasks.

这些图具体怎么生成呢？

- Power map:Layout information提供了电路布局中每个实例和块的位置，Power database提供了充分的功率分析结果，二者结合就能够绘制出二维的功率分布图Power map.
- PDN density map:通过提取芯片每一个区域的平均PDN密度生成。首先将PDN分区域统计线路密度，具体方式论文没有说，应该是统计每个区域Block,stripes等的数量或者规模，然后算出单位面积的规模作为密度。基于此绘制出density map.
- Effective distance to pad:PDN上有多个电源焊盘，针对芯片上的所有实例，计算到所有这些焊盘的等效距离，作为它的effective distance。这个距离计算为实例到各个电源焊盘的调和和：

$$d_e^{-1} = d_1^{-1} + d_2^{-1} + \dots + d_N^{-1}$$

然后根据每一个点距离远近上色，就得到了上图Fig2(f)的效果。这个距离跟Power map共同表征了实例与焊盘之间的等效电阻。这项研究直接使用这个距离来充当电阻。

在完成这些工作之后，数据就能够投入到模型中进行训练了。因为我们做的是静态分析，所以要结合三张图片输入到模型中进行训练，完成一个图像翻译任务。这里使用的网络架构是**基于U-Net的EDGE网络**。它的输出是整张芯片压降的热力图。

3.模型与算法

基于U-Net的EDGE网络是一种特定类型的神经网络架构，已被证明在图像翻译任务中非常有效。

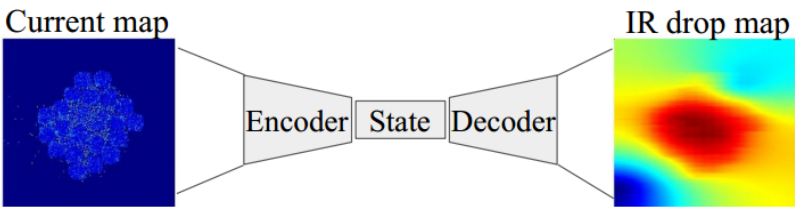


Fig. 1. Image-to-image translation using EDGe network.

U-Net架构具有对称的编码器-解码器结构。编码器部分从输入功率图像中提取高级特征，而解码器部分生成具有所需特征的输出图像，例如温度分布或IR压降轮廓。该网络是在数据集上进行训练的，该数据集包含一组输入功率图像和相应的真实输

出图像，这些真实输出图像通常是通过物理仿真或测量生成的。

通过在大量功率图像及其相应的轮廓图或梯度图数据集上训练基于U-Net的EDGE网络，网络学习捕捉功率分布与温度或IR压降分布之间的潜在模式和关系。一旦训练完成，该网络可以用于预测训练集中未包含的新输入功率的IR压降分布。

这个网络结构如下图所示。

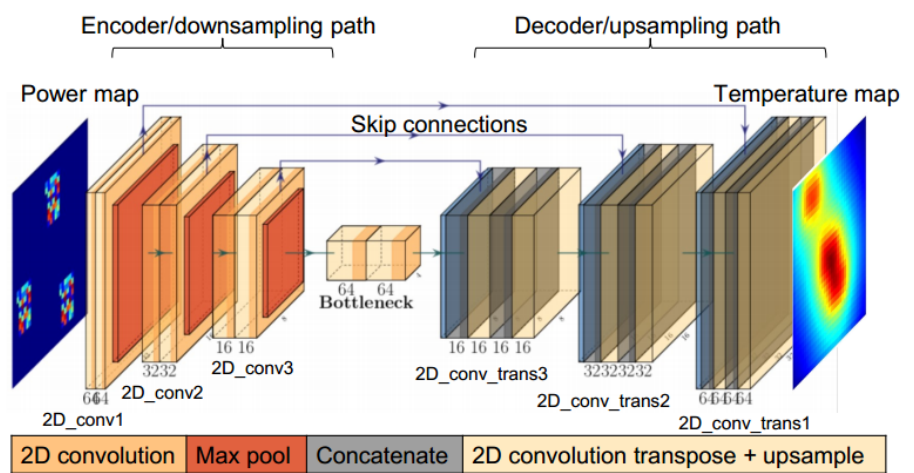


Fig. 3. U-Net-based EDGE network for static thermal and PDN analysis.

模型概述

在顶层，它由两个网络组成：

a.编码器/下采样网络(Encoder/Downsampling)

类似于CNN，该网络利用一系列的2D卷积和最大池化层对高维度输入特征集进行关键特征提取。卷积操作在图像上的滑动窗口上进行加权求和，而Max pool通过在输入图像上的滑动窗口中提取最大值来减小输入数据的维度。在图3中，每个层对通过每个层对的操作将特征维度减半，经过多次这样的操作后，就得到编码的、低维度的、压缩的输入数据表示。这个过程叫做下采样。

这样做的目的跟池化操作很像，因为这实现了特征的汇聚，有利于集中各种区域信息，例如，某一块区域是否有电压降很大的点，等等；但是下采样也让确切的位置信息变得不够精确了，所以还需要上采样（解码器）来恢复。

b.解码器/上采样网络(Decoder/Upsampling)

生成解码器负责检索在下采样过程中丢失的“位置”信息。解码器使用反卷积和上采样层进行实现。上采样层在功能上与池化层相反，通过不断倍增行和列来增加输入数据矩阵的维度。

反卷积(Transpose Convolution)是卷积操作的反变换，旨在将低维的特征转换为高维的特征。详情参考[这篇博客](#)。

跳跃连接(skip connections)

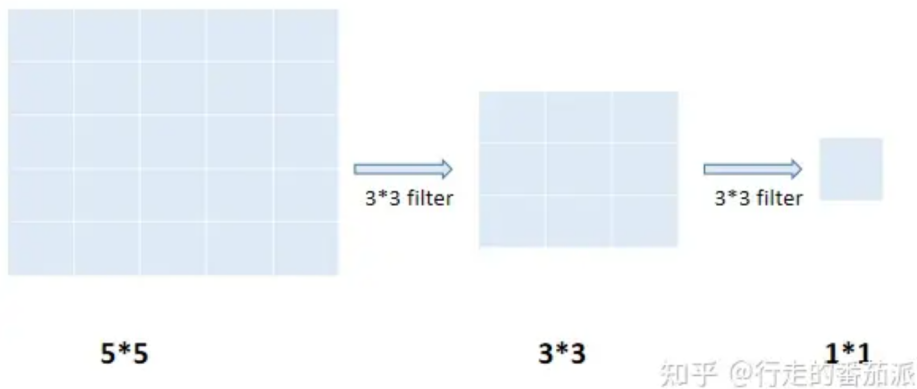
这个网络的鲜明特征是，使用了跳跃连接的技术(skip connections):将下采样与上采样路径之间的特征图用一种“连接层(concatenation layer)”跳跃中间层直接连接起来，连接方式是直接堆叠在最后一维(z维，看图即可)。注意，每一次跳跃连接都是下采样阶段**刚刚执行完卷积操作之后的特征图**与上采样阶段反卷积层的输入相连，并不是执行卷积之前的图。这一点要仔细观察。

之所以这么做，还是因为要逐步恢复下采样阶段丢失的准确位置信息，从而“防止模型的生成跑偏”。

感受野

感受野(receptive field)是一个抽象的概念，表示卷积神经网络每一层输出的特征图（feature map）上的像素点映射回原始输入图像上的区域大小。这跟卷积核与池化操作有密切的关系。像下面这个图，我输入了5*5的矩阵到神经网络，它经历两次卷

积最后缩到1*1的点，那么这个点的感受野就是5*5。



不难发现，随着下采样的逐渐加深，每一个点的感受野会越来越大。

科学家发现，在深度网络中，每个像素特征的值受到输入图像的感受野中所有像素的影响，其中最大的贡献来自感受野中心附近的像素。因此，每个特征不仅捕捉输入图像中的感受野，而且对该区域的中心部分给予指数级更高的权重。

这也进一步佐证了为什么输出的图像能够放心地认为是实际的压降分布图。因为经历了上采样之后，每一个像素点依旧与它感受野中心的那个点是最相关的。而恰好这两个点在输入和输出图中代表同一个位置的像素。

4.模型的训练和评估

模型的训练

我们可以参考论文中的模型训练模式进行训练。

TABLE I
THERMEDGE AND IREDGE ML HYPERPARAMETERS

ML hyperparameters			Static ThermEDGe	IREdGe	Transient ThermEDGe
Model layer parameters	2D_conv1	filter size	5x5	3x3	5x5
	2D_conv_trans1	# filters	64	64	64
	2D_conv2	filter size	3x3	3x3	3x3
	2D_conv_trans2	# filters	32	32	32
	2D_conv3	filter size	3x3	3x3	–
	2D_conv_trans3	# filters	16	16	–
	Max pool layers	filter size	2x2	2x2	2x2
	ConvLSTM	filter size	–	–	7x7
Training parameters			–	–	16
	Epochs		500		
	Optimizer		ADAM		
	Loss function		Pixelwise MSE		
	Decay rate		0.98		
	Decap steps		1000		
	Regularizer		L2		
	Regularization rate		1.00E-05		
	Learning rate		1.00E-03		

模型的评估

论文使用的是图级别的误差评估。使用绝对误差

$$IR_{err} = |IR_{true} - IR_{pred}|$$

其中 IR_{true} 是使用商业分析软件得到的结果图像，可以认为是正确的。而 IR_{pred} 是使用模型预测的IR分布图。误差是两个图像像素级作差求绝对值之后取全局平均或者全局最大值。

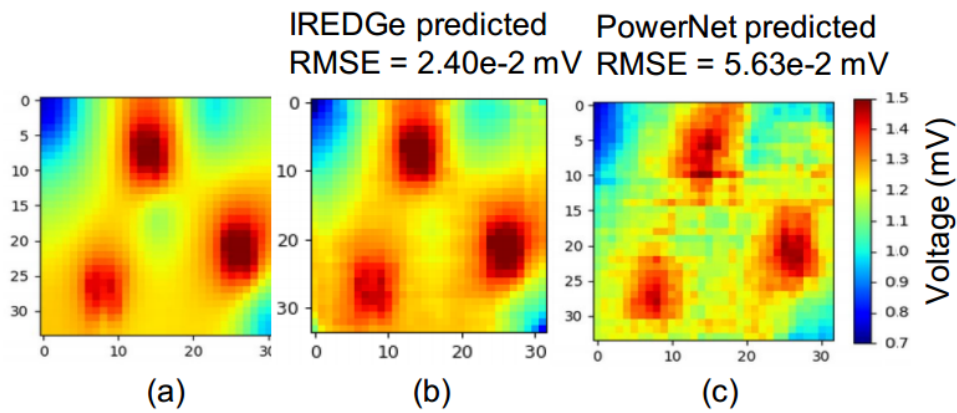
模型比powernet要好

论文对两个模型的推理时间做了对比实验：

芯片尺寸	Powernet	IREEDGE
34*32	3.2ms	1.1ms
68*32	6.2ms	1.3ms

这也从侧面反映出，powernet在原理上占劣势。因为它得先把电路分tile，然后一个一个tile这样进行预测，所以可以看出芯片规模一大，它的推理时间也翻倍了。而IREEDGE因为不涉及尺寸，只进行一次预测行为，只是做卷积等事情时稍微多占了一些时间，所以总体性能并没有怎么下降。

此外，IREEDGE的效果也比Powernet更好。在相同的测试集下，训练好的IREEDGE平均误差是0.028mV,最大误差是0.14mV，而powernet平均误差是0.042mV，最大误差是0.17mV。下图先后展现了标准结果、IREEDGE分析结果和powernet分析结果。



单从图像的相似程度上，我们也能看出IREEDGE更胜一筹。powernet之所以显得比较凌乱，应该也是由于它是分块预测所导致的。