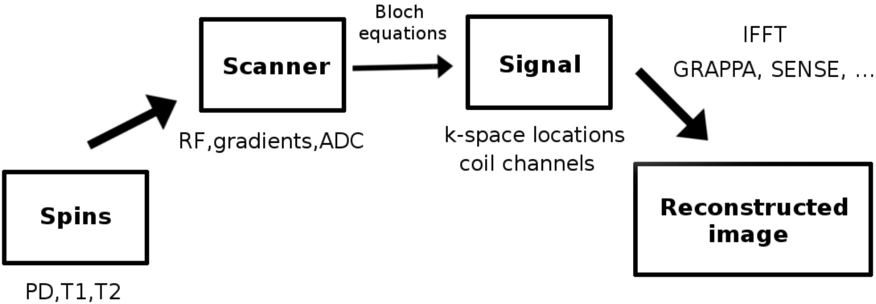
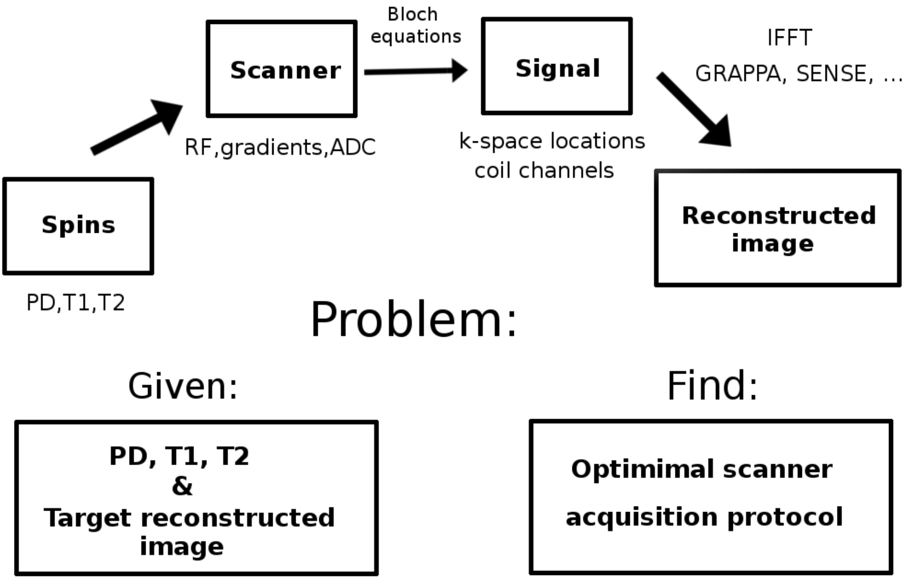
Big picture



Bloch
equations
Scanner
Signal
IFFT
GRAPPA, SENSE, ...
RF,gradients,ADC
k-space locations
coil channels
Spins
Reconstructed
image
PD,T1,T2

Problem definition



Formulate as a machine learning supervised problem

Input:
Initial magnetization tensor: $\mathbf{m} \in \mathbb{R}^{Nsamples \times Nspins \times Nvoxels \times 3}$

Predict:
Target reconstructed image tensor: $\mathbf{u} \in \mathbb{R}^{Nsamples \times Nvoxels \times 2}$

Formulate as a machine learning supervised problem

Input:
Initial magnetization tensor: $\mathbf{m} \in \mathbb{R}^{Nsamples \times Nspins \times Nvoxels \times 3}$

Predict:
Target reconstructed image tensor: $\mathbf{u} \in \mathbb{R}^{Nsamples \times Nvoxels \times 2}$

Forward model:
$\mathbf{u} = RECO_{\mathbf{\Theta}}(SCANNER_{\mathbf{\Phi}}(\mathbf{m}))$

Formulate as a machine learning supervised problem

Input:
Initial magnetization tensor: $\mathbf{m} \in \mathbb{R}^{Nsamples \times Nspins \times Nvoxels \times 3}$

Predict:
Target reconstructed image tensor: $\mathbf{u} \in \mathbb{R}^{Nsamples \times Nvoxels \times 2}$

Forward model:
$\mathbf{u} = RECO_{\mathbf{\Theta}}(SCANNER_{\mathbf{\Phi}}(\mathbf{m}))$

SCANNER - scanner function
RECO - reconstruction function

Formulate as a machine learning supervised problem

Input:
Initial magnetization tensor: $\mathbf{m} \in \mathbb{R}^{Nsamples \times Nspins \times Nvoxels \times 3}$

Predict:
Target reconstructed image tensor: $\mathbf{u} \in \mathbb{R}^{Nsamples \times Nvoxels \times 2}$

Forward model:
$\mathbf{u} = RECO_{\mathbf{\Theta}}(SCANNER_{\mathbf{\Phi}}(\mathbf{m}))$

SCANNER - scanner function
RECO - reconstruction function

Learn sequence and reconstruction module parameters jointly:
$\mathbf{\Theta^*}, \mathbf{\Phi}^* = \arg\min_{\mathbf{\Theta}, \mathbf{\Phi}}(\|\mathbf{u} - RECO_{\mathbf{\Theta}}(SCANNER_{\mathbf{\Phi}}(\mathbf{m}))\|_2^2)$

Scanner function parameters

$\mathbf{\Phi} = \{\boldsymbol{\alpha}, \mathbf{\Delta t}, \mathbf{g_{x,y}}, \mathbf{a}\}$

flip angle tensor $\boldsymbol{\alpha} \in [-\pi..\pi]^{Nrepetitions \times Nactions}$

time delay (relaxation) tensor $\mathbf{\Delta t} \in \{\mathbb{R}^+\}^{Nrepetitions \times Nactions}$

gradient moment tensor $\mathbf{g_{x,y}} \in \mathbb{R}^{Nrepetitions \times Nactions \times 2}$

adc switcher tensor $\mathbf{a} \in \{0, 1\}^{Nrepetitions \times Nactions}$

Forward process (single sample)

Initialize: $\mathbf{m}_0$ to initial magnetization state

for $r \leftarrow 0 : N_{repetitions}$ **do**

  for $a \leftarrow 0 : N_{actions}$ **do**

    $i = r * N_{actions} + a$

    $\mathbf{m}_i = FLIP_y(\alpha_i)\mathbf{m}_i$

    $\mathbf{m}_i = RELAX(\Delta t_i)\mathbf{m}_i$

    $\mathbf{m}_i = FREEPRECESS(\Delta t_i)\mathbf{m}_i$

    $\mathbf{m}_i = GRADPRECESS(g_{x,i} + g_{y,i})\mathbf{m}_i$

    $\mathbf{s}_i = \mathbf{a}_i \sum_{Nvoxels,NSpins} \mathbf{m}_{i,transverse}$

    $\mathbf{m}_{i+1} = \mathbf{m}_i$

  end

end

Output: signal $\mathbf{s} \in \mathbb{R}^{Nrepetitions \times Nactions \times 2}$

Modeling the acquisition

RF operator:

$$FLIP_y(\alpha_i) = \begin{pmatrix} cos(\alpha_i) & 0 & -sin(\alpha_i) \\ 0 & 1 & 0 \\ sin(\alpha_i) & 0 & cos(\alpha_i) \end{pmatrix}$$

Relaxation operator:

$$RELAX(\Delta t_i) = \begin{pmatrix} exp(\frac{-\Delta t_i}{T2}) & 0 & 0 \\ 0 & exp(\frac{-\Delta t_i}{T2}) & 0 \\ 0 & 0 & exp(\frac{-\Delta t_i}{T1}) \end{pmatrix}$$

Gradient precession operator:

$$GRADPRECESS(g_i) = \begin{pmatrix} cos(g_i) & -sin(g_i) & 0 \\ sin(g_i) & cos(g_i) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Free precession operator:

$$FREEPRECESS(\Delta t_i) = \begin{pmatrix} cos(\Delta t_i \omega_{off}) & -sin(\Delta t_i \omega_{off}) & 0 \\ sin(\Delta t_i \omega_{off}) & cos(\Delta t_i \omega_{off}) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Scanner as a linear operator

- Since the forward process involves a (potentially very long) chain of tensor-tensor multiplications, it can be represented as a single tensor multiplication with initial magnetization vector:

  $\mathit{signal} = \mathbf{S}_{\{\boldsymbol{\alpha}, \boldsymbol{\Delta}\mathbf{t}, \mathbf{g}_{\mathbf{x},\mathbf{y}}, \mathbf{a}\}} \mathbf{m}_0$, where $\mathbf{S}$ is a scanner tensor.

- Deep learning libraries such as Tensorflow, Torch, Caffee are all very efficient at handling such massive parallel tensor operations. Using graphics card for this purpose can reduce computation time by orders of magnitude.

- Numerical derivatives of the optimized variables can be found very efficiently and with minimum implementation time overhead by means of autodifferentiationm which Deep learning packages above implement.

Reconstruction function

Learn sequence and reconstruction module parameters jointly:

$$\boldsymbol{\Theta}^{*}, \boldsymbol{\Phi}^{*} = \underset{\boldsymbol{\Theta},\boldsymbol{\Phi}}{\arg\min}(\|\mathbf{u} - RECO_{\boldsymbol{\Theta}}(SCANNER_{\boldsymbol{\Phi}}(\mathbf{m}))\|_2^2)$$

Reconstruction function

Consider for now only spatial encoding gradient-driven precession:. Also assume we deal only with a transverse component of the magnetization vector $\boldsymbol{m}$.

$$\boldsymbol{s}(t) = \int_V \boldsymbol{m}(\boldsymbol{r}, t_1) * exp(i(g_x(t)ramp_x(\boldsymbol{r}) + g_y(t)ramp_y(\boldsymbol{r}))dt)dxdydz$$

Reconstruction function

Consider for now only spatial encoding gradient-driven precession:. Also assume we deal only with a transverse component of the magnetization vector $\boldsymbol{m}$.

$$\boldsymbol{s}(t) = \int_V \boldsymbol{m}(\boldsymbol{r}, t_1) * exp(i(g_x(t) ramp_x(\boldsymbol{r}) + g_y(t) ramp_y(\boldsymbol{r}))dt) dxdydz$$

Matrix form:
$\boldsymbol{s} = \mathbf{E}_{g_{x,y}} \boldsymbol{m}$

Closed form inverse solution:
$\hat{\boldsymbol{m}} \approx \mathbf{E}^H_{g_{x,y}} \boldsymbol{s}$

Reconstruction function

Consider for now only spatial encoding gradient-driven precession:. Also assume we deal only with a transverse component of the magnetization vector $\boldsymbol{m}$.

$$\boldsymbol{s}(t) = \int_V \boldsymbol{m}(\boldsymbol{r}, t_1) * exp(i(g_x(t)ramp_x(\boldsymbol{r}) + g_y(t)ramp_y(\boldsymbol{r}))dt)dxdydz$$

Matrix form:

$\boldsymbol{s} = \mathbf{E}_{g_{x,y}}\boldsymbol{m}$

Closed form inverse solution, $\mathbf{E}^H_{g_{x,y}}$ is an adjoint of encoding operator:

$\hat{\boldsymbol{m}} \approx \mathbf{E}^H_{g_{x,y}}\boldsymbol{s}$

Objective function:

$$\mathbf{g_{x,y}}^* = \arg\min_{\mathbf{g_{x,y}}}(\left\|\mathbf{m} - \mathbf{E}^H_{g_{x,y}}\mathbf{E}_{g_{x,y}}\mathbf{m})\right\|_2^2)$$

Reconstruction function

Assume repetitions with a very long TR, and a single RF event inside each repetition. To solve for scanner parameters we can optimize:

$$\mathbf{\Phi}^* = \arg\min_{\mathbf{\Phi}} \left(\left\| \mathbf{u} - \mathbf{E}_{g_{x,y}}^{H}(SCANNER_{\mathbf{\Phi}}(\mathbf{m})) \right\|_2^2 \right)$$

where $\mathbf{E}_{g_{x,y}} \in \mathbf{\Phi}$

Multiple echo case: convolutional NN

Using adjoint is no longer valid, since each echo will mix k-space locations from the older echoes.

$$\boldsymbol{\Theta}^*, \boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\Phi}} \left(\left\| \mathbf{u} - NN_{\boldsymbol{\Theta}}(\mathbf{E}^H_{g_{x,y}}(SCANNER_{\boldsymbol{\Phi}}(\mathbf{m}))) \right\|_2^2 \right)$$

where $NN_{\boldsymbol{\Theta}}$ is a fully convoltional network, with parameters $\boldsymbol{\Theta}$ describing convolution weights.

Tissue characterization

Relax the requirement of having a fixed target from a given sequence. Predict tissue parameters directly.

$$\boldsymbol{\Theta}^{*}, \boldsymbol{\Phi}^{*} = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\Phi}} \left(\left\| \{PD, T1, T2\} - NN_{\boldsymbol{\Theta}}(\mathbf{E}^{H}_{g_{x,y}}(SCANNER_{\boldsymbol{\Phi}}(\mathbf{m}))) \right\|_2^2 \right)$$